Лекція 12. Аутентифікація інформації та функції хешування (частина 1).

1. Функции аутентификации

В аутентификации сообщений и средствах цифровой подписи могут быть выделены два основных уровня. На низшем уровне должна выполняться некоторая функция, порождающая аутентификатор (удостоверение) – значение, используемое для аутентификации сообщения. Результат выполнения этой функции низшего уровня затем используется как примитив в протоколе аутентификации высшего уровня, дающем получателю сообщения возможность проверить достоверность сообщения.

В этом разделе мы рассмотрим типы функций, которые могут служить для создания аутентификатора. Такие функции можно разделить на три следующих класса.

- **Шифрование сообщения.** В качестве аутентификатора используется шифрованный текст всего сообщения.
- **Код аутентичности сообщения (Message Authentication Code – MAC).** В качестве аутентификатора выступает значение фиксированной длины, генерируемое некоторой открытой функцией сообщения и секретным ключом.
- **Функция хэширования.** В качестве аутентификатора используется значение, генерируемое некоторой открытой функцией, противопоставляющей любому сообщению произвольной длины значение фиксированной длины, называемое значением хэш-функции.

Теперь рассмотрим каждую из этих возможностей вкратце, а коды аутентичности сообщений и функции хэширования будут описаны подробнее.

1.1. Шифрование сообщения

Шифрование сообщения само по себе в некоторой мере может выполнять функции аутентификации сообщения. Анализ схемы для традиционного шифрования и шифрования с открытым ключом оказывается при этом различным.

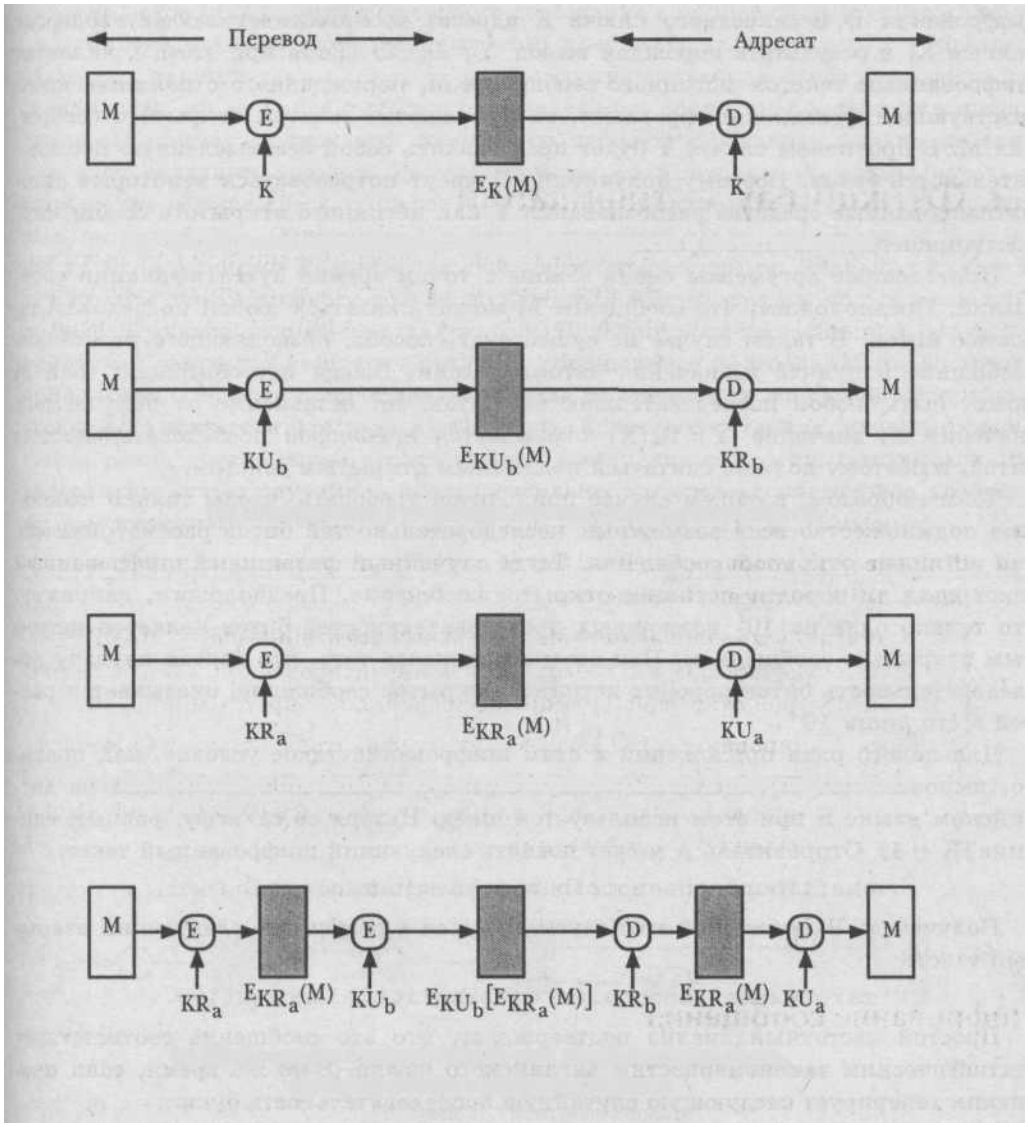
1.1.1. Традиционное шифрование

Рассмотрим непосредственное использование традиционного шифрования (рис. 8.1(а)). Сообщение, передаваемое от источника **A** адресату **B**, шифруется с использованием секретного ключа K, известного обеим сторонам, **A** и **B**. Если никто другой не знает ключа, конфиденциальность гарантируется – никто другой не сможет восстановить открытый текст сообщения.

Кроме того, можно утверждать, что пользователь **B** будет уверен в том, что отправителем пришедшего сообщения является пользователь **A**. Почему? Сообщение должно исходить от пользователя **A** ввиду того, что кроме пользователя **B** только пользователь **A** владеет ключом K и поэтому никто другой не располагает информацией, необходимой для создания шифрованного текста, который может быть дешифрован с помощью K. Кроме того, если сообщение M восстанавливается, это доказывает получателю **B**, что никакой из битов M не был изменен, так как противник, не обладающий ключом K, не сможет изменить биты в шифрованном тексте так, чтобы произвести желаемые изменения в открытом тексте сообщения.

Таким образом, можно сказать, что традиционное шифрование обеспечивает аутентификацию сообщений, а не только их конфиденциальность. Однако это простое утверждение все же требует уточнения. Рассмотрим, например, что конкретно происходит с точки зрения пользователя **B**. При наличии функции дешифрования D и секретного ключа K адресат воспринимает *любые* вводимые данные X, в результате порождая вывод $Y = D_K(X)$. Если при этом X является шифрованным текстом истинного сообщения M, порожденного с помощью соответствующей функции шифрования, то Y окажется текстом открытого сообщения M. В противном случае Y будет представлять собой бессмысленную последовательность битов. Поэтому получателю **B** могут потребоваться некоторые автоматизированные средства распознавания Y как истинного открытого сообщения, поступившего от инициатора **A**.

Приведенные аргументы очень важны с точки зрения аутентификации сообщений. Предположим, что сообщение M может оказаться любой последовательностью битов. В таком случае не существует способа, позволяющего распознать сообщение в пункте назначения автоматически. Вывод неутешителен: если M может быть любой последовательностью битов, то, независимо от получаемого значения X, значение $Y = D_K(X)$ оказывается *некоторой* последовательностью битов, и поэтому должно считаться подлинным открытым текстом.



(а) Традиционное шифрование: конфиденциальность и аутентификация
(б) Шифрование с открытым ключом: конфиденциальность
(в) Шифрование с открытым ключом: конфиденциальность и цифровая подпись
(г) Шифрование с открытым ключом: конфиденциальность, аутентификация и цифровая подпись

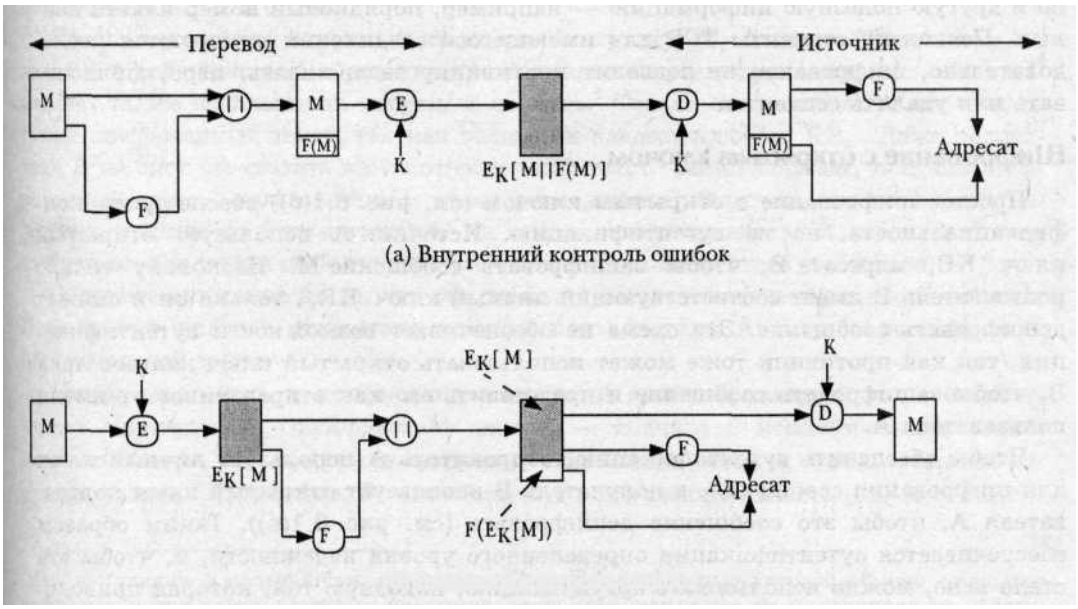
Рис. 8.1. Основные возможности шифрования сообщений.

Таким образом, в общем случае приходится требовать, чтобы только небольшое подмножество всех возможных последовательностей битов рассматривалось как истинные открытые сообщения. Тогда случайный фальшивый шифрованный текст вряд ли породит истинное открытое сообщение. Предположим, например, что только одна из 10^6 возможных последовательностей битов является истинным открытым сообщением. При этом вероятность того, что взятая наудачу последовательность битов породит истинное открытое сообщение, оказывается равной всего лишь 10^{-6}.

Иногда бывает трудно определить *автоматически,* дешифруется ли поступающий шифрованный текст в *понятный* открытый текст. Если же открытый текст является, скажем, двоичным объектным файлом или цифровой записью рентгеновского излучения, определение правильности дешифрования и подлинности получаемого таким образом открытого текста может оказаться весьма трудным делом. В таком случае противник может выдать за сообщение от законного источника практически любое сообщение с произвольным содержимым.

Одной из возможностей решения этой проблемы является структурирование открытых текстов таким образом, чтобы соответствующая структура легко распознавалась, но не могла быть воспроизведена без обращения к функции шифрования. Например, к каждому сообщению перед шифрованием можно добавить код распознавания ошибок, называемый также контрольной последовательностью кадра (Frame check sequence – FCS), или контрольной суммой, как показано на рис. 8.2(а). Отправитель **А** берет открытый текст сообщения M и подает его на вход функции F, вычисляющей контрольную сумму. Значение контрольной суммы присоединяется к M, и полученный таким образом новый блок шифруется. В пункте назначения получатель **В** дешифрует поступивший блок, рассматривая результат дешифрования как сообщение с присоединенной контрольной суммой.

Поэтому получатель **B** сначала применяет ту же самую функцию F, чтобы воспроизвести контрольную сумму. Если подсчитанная им контрольная сумма равна поступившей с сообщением, сообщение считается подлинным. Маловероятно, чтобы случайная последовательность битов удовлетворяла требуемому соотношению.



(б) Внешний контроль ошибок

Рис. 8.2. Внутренний и внешний контроль ошибок

Обратите внимание на порядок, в котором выполняются вычисление контрольной суммы и шифрование. Последовательность выполнения этих операций, показанная в виде схемы на рис. 8.2(а), называется внутренним контролем ошибок, который авторы противопоставляют внешнему контролю ошибок (рис. 8.2(б)). В случае внутреннего контроля ошибок обеспечивается и аутентификация, поскольку противник сталкивается с проблемой генерирования шифрованного текста, дающего при дешифровании правильное значение контрольной суммы. Если вместо внутренней использовать внешние контрольные суммы, противник получает возможность создавать сообщения с правильными кодами контроля ошибок. И хотя при этом ему не удастся получить открытый текст сообщения, он сможет создавать ложные и искаженные сообщения.

Коды контроля ошибок являются только одним примером – на самом деле любая структуризация передаваемых сообщений расширяет возможности аутентификации. Такая структуризация обеспечивается архитектурой коммуникаций, формируемой из набора используемых при этом протоколов передачи данных.

1.1.2. Шифрование с открытым ключом.

Простое шифрование с открытым ключом (см. рис. 8.1(б)) обеспечивает конфиденциальность, но не аутентификацию. Источник **A** использует открытый ключ KU_B адресата **B**, чтобы зашифровать сообщение M. Поскольку только пользователь **B** имеет соответствующий личный ключ KR_B, только он и сможет дешифровать сообщение. Эта схема не обеспечивает возможности аутентификации, так как противник тоже может использовать открытый ключ пользователя **B**, чтобы зашифровать сообщение и представить его как отправленное от имени пользователя **A**.

Чтобы обеспечить аутентификацию, отправитель **A** использует личный ключ для шифрования сообщения, а получатель **B** использует открытый ключ пользователя **A**, чтобы это сообщение дешифровать (см. рис. 8.1(в)). Таким образом обеспечивается аутентификация определенного уровня надежности, и, чтобы это стало ясно, можно использовать аргументацию, подобную той, которая приводилась ранее для случая традиционного шифрования. Ее суть состоит в том, что отправителем сообщения должен быть пользователь **A**, поскольку только он имеет KR_A и, следовательно, только пользователь **A** обладает информацией, необходимой для создания шифрованного текста, который может быть дешифрован с помощью KU_A. Здесь, как и в предыдущем случае, возникает проблема: открытый текст должен обладать некоторой внутренней структурой, чтобы получатель мог отличить явно структурированный открытый текст от случайной последовательности битов.

Предположим, что открытый текст такой структурой обладает. Тогда схема на рис. 8.1 (в) действительно обеспечивает возможность аутентификации. Эта схема позволяет также использовать цифровую подпись. (Как мы увидим в дальнейшем, реальная процедура, с помощью которой обычно создаются цифровые подписи, отличается от представленного здесь описания, но принципиально

является точно такой.) Только пользователь **A** мог создать такой шифрованный текст, так как только он владеет ключом KR_A. Даже получатель **B** не смог бы создать этот шифрованный текст. Таким образом, получив шифрованный текст, пользователь **B** имеет возможность убедиться, что сообщение наверняка пришло от отправителя **A**. На самом деле отправитель **A** "подписал" свое сообщение, используя свой личный ключ для шифрования этого сообщения.

Обратите внимание на то, что эта схема не обеспечивает конфиденциальности. Любой, кто владеет открытым ключом пользователя **A**, может дешифровать шифрованный текст.

Чтобы обеспечить и конфиденциальность, и аутентификацию, отправитель **A** может зашифровать сообщение M дважды – сначала с использованием своего личного ключа, что обеспечит цифровую подпись, а затем с использованием открытого ключа получателя **B**, что обеспечит конфиденциальность (см. рис. 8.1(г)).

Недостаток такого подхода состоит в том, что алгоритм шифрования, который в схеме с открытым ключом является весьма сложным, должен в каждом сеансе связи применяться четыре раза, а не два.

Информация о степени конфиденциальности и аутентификации, обеспечиваемой при использовании разных подходов к шифрованию сообщений, представлена в табл. 8.1.

Таблица 8.1. Конфиденциальность и аутентификация сообщений при шифровании

(а) Традиционное (симметричное) шифрование
$A \rightarrow B:\ \ E_K[M]$ • Обеспечивает конфиденциальность - Только стороны **A** и **B** знают K • Обеспечивает определенный уровень аутентификации - Источником может быть только сторона **A** - Изменения в пути следования невозможны - Требуется форматирование/избыточность • Не обеспечивает подпись - Получатель имеет возможность фальсифицировать получение сообщения - Отправитель имеет возможность отрицать отправку сообщения
(б) Шифрование с открытым ключом (асимметричное шифрование)
$A \rightarrow B:\ \ E_{KUb}[M]$ • Обеспечивает конфиденциальность - Только сторона **B** имеет KR_B, чтобы дешифровать сообщение • Не обеспечивает аутентификацию - Кто угодно может использовать KU_B для шифрования сообщения, чтобы объявить себя отправителем **A**
$A \rightarrow B:\ \ E_{KRa}[M]$ • Обеспечивает аутентификацию и подпись - Только сторона **A** имеет KR_A, чтобы зашифровать сообщение - Изменения на пути следования невозможны - Требуется форматирование/избыточность - Кто угодно может использовать KU_A, чтобы проверить подпись
$A \rightarrow B:\ \ E_{KUb}[E_{KRa}[M]]$ • Обеспечивает конфиденциальность, поскольку используется KU_B • Обеспечивает аутентификацию и подпись, поскольку используется KR_A

1.2. Код аутентичности сообщения

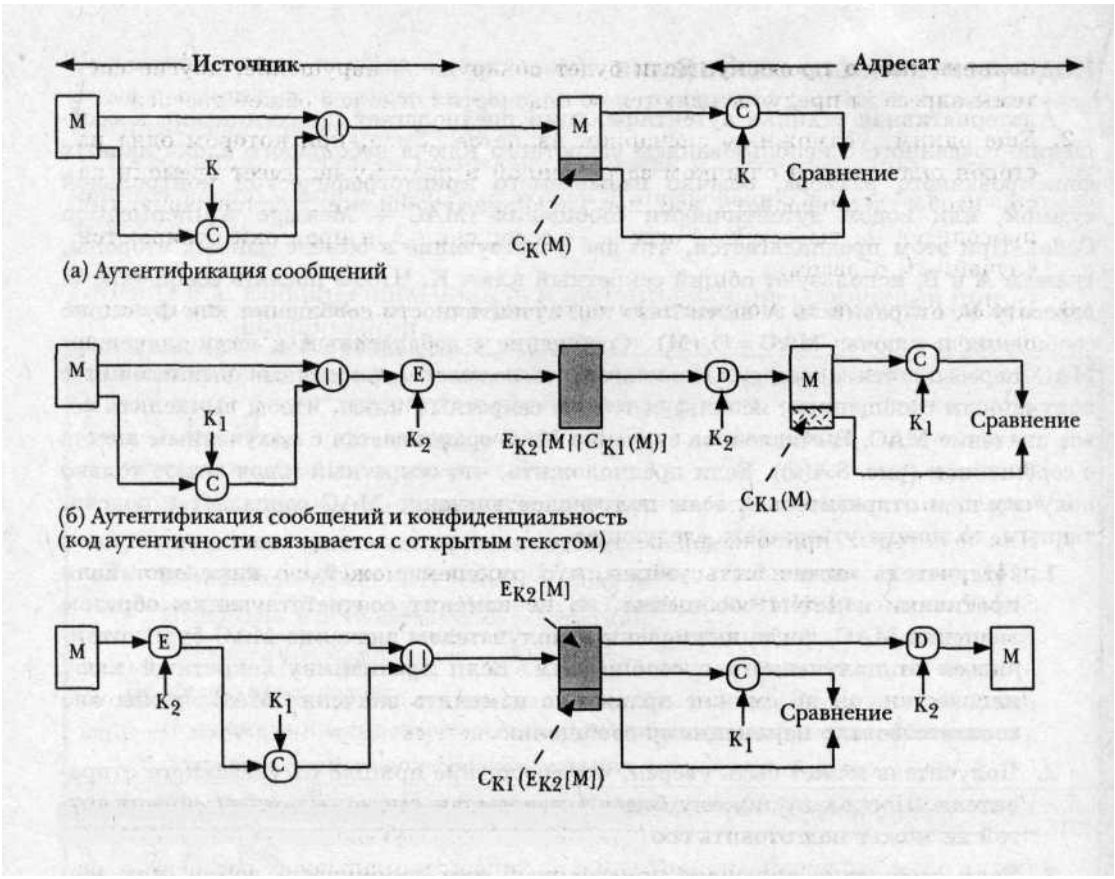
Альтернативная техника аутентификации предполагает присоединение к сообщению созданного с использованием секретного ключа небольшого блока данных фиксированного размера, обычно называемого криптографической контрольной суммой, или кодом аутентичности сообщения (MAC – Message Authentication Code). При этом предполагается, что две участвующие в обмене данных стороны, скажем **A** и **B**, используют общий секретный ключ K. Чтобы послать сообщение M адресату **B**, отправитель **A** вычисляет код аутентичности сообщения как функцию сообщения и ключа:

$MAC = C_K(M)$. Сообщение с добавленным к нему значением MAC пересылается адресату. Получатель выполняет аналогичные вычисления с полученным сообщением, используя тот же секретный ключ, чтобы вычислить новое значение MAC. Вычисленное значение MAC сравнивается с полученным вместе с сообщением (рис. 8.4(а)). Если предположить, что секретный ключ знают только получатель и отправитель и если полученное значение MAC совпадает с подсчитанным, то можно утверждать следующее.

1. Получатель может быть уверен, что сообщение не было изменено. Если противник изменит сообщение, но не изменит соответствующим образом значение MAC, тогда вычисленное получателем значение MAC будет отличаться от полученного с сообщением. Если противнику секретный ключ неизвестен, он не сможет правильно изменить значение MAC, чтобы оно соответствовало измененному сообщению.
2. Получатель может быть уверен, что сообщение пришло от указанного отправителя. Поскольку никому больше неизвестен секретный ключ, никто другой не может подготовить сообщение с соответствующим значением MAC.
3. Если сообщение включает присвоенный ему порядковый номер (как это, например, предполагается в случае использования протоколов HDLC, X.25 или TCP), то получатель может быть уверен в том, что сообщения приходят в правильной последовательности, поскольку противник не имеет возможности изменить порядковый номер сообщения.

Функция вычисления MAC подобна по своим действиям шифрованию. Разница лишь в том, что для алгоритма вычисления MAC не требуется свойство обратимости, как для алгоритма шифрования, чтобы можно было дешифровать сообщение. Оказывается, что такие математические свойства функции аутентификации делают ее менее уязвимой ко взлому по сравнению с функцией шифрования.

Только что описанный процесс обеспечивает аутентификацию, но не конфиденциальность, поскольку сообщение в целом передается открытым. Конфиденциальность может быть обеспечена путем шифрования сообщения либо после (рис. 8.4(б)), либо перед (рис. 8.4(в)) применением алгоритма вычисления MAC. В любом из этих случаев потребуется два ключа, известные и отправителю, и получателю. В первом случае значение MAC вычисляется для подающегося на вход сообщения и результат присоединяется к сообщению путем конкатенации. Затем весь блок шифруется. Во втором случае сначала шифруется сообщение, затем вычисляется значение MAC для шифрованного текста и результат присоединяется путем конкатенации к шифрованному сообщению, чтобы сформировать передаваемый блок. Обычно предпочтительнее связывать код аутентичности непосредственно с открытым текстом, поэтому чаще применяется метод, схема которого представлена на рис. 8.4(б).



(в) Аутентификация сообщений и конфиденциальность (код аутентичности связывается с шифрованным текстом)

Рис. 8.4. Основные способы использования кодов аутентичности сообщений (MAC).

Поскольку традиционное шифрование обеспечивает аутентификацию и поскольку такое шифрование можно осуществить с помощью уже доступных средств, находящихся в широком использовании, почему бы не выбрать именно этот способ, вместо того чтобы вводить новый код аутентичности сообщения? Предлагается рассматривать следующие три ситуации, в которых имеет смысл использовать коды аутентичности сообщений.

1. Имеется целый ряд приложений, в которых одно и то же сообщение может передаваться сразу нескольким адресатам. Примерами могут служить извещение пользователей о том, что сеть в данный момент недоступна, или сигнал тревоги, посылаемый из центрального военного штаба. Дешевле и надежнее иметь одного адресата, ответственного за проверку аутентичности. При таком подходе сообщение должно передаваться в открытом виде вместе с соответствующим кодом аутентичности сообщения. Ответственная за проверку аутентичности система должна иметь секретный ключ и выполнить такую проверку. Если будет обнаружено нарушение, другие системы-адресаты предупреждаются об опасности сигналом общей тревоги.
2. Еще одним возможным сценарием является обмен, при котором одна из сторон оказывается слишком загруженной и поэтому не имеет времени на то, чтобы дешифровать все поступающие сообщения. Аутентификация проводится на выборочной основе – сообщения для проверки отбираются случайным образом.
3. Привлекательна возможность аутентификации компьютерных программ в нешифрованном виде. Компьютерная программа может быть выполнена и без того, чтобы каждый раз дешифровать ее, что требует дополнительных ресурсов процессора. Если с такой программой связать код аутентичности, то его можно проверять каждый раз, когда потребуется проверить целостность программы.

К этим вариантам можно добавить три следующих.

1. Для некоторых приложений не требуется сохранять секретность, но важно проверить аутентичность сообщений. Примером может служить протокол SNMP версии 3 (SNMPv3 – Simple Network Management Protocol Version 3), где функции конфиденциальности и аутентификации разделяются. В этом приложении для управляемой системы важно гарантировать аутентичность поступающих SNMP-сообщений, особенно, если сообщение содержит команды, изменяющие параметры управляемой системы. В то же время необходимость скрывать поток обмена данными SNMP может не требоваться.
2. Разделение функций аутентификации и конфиденциальности обеспечивает повышение гибкости архитектуры. Например, можно осуществить аутентификацию на уровне приложения, а конфиденциальность обеспечить на более низком уровне, например на транспортном.
3. Пользователь может пожелать оставить сообщение защищенным и после его получения, но при этом иметь возможность ознакомиться с его содержимым. После дешифрования сообщения защита не гарантируется, так что сообщение оказывается защищенным от модификаций только на пути следования, но не в самой системе назначения.

Наконец, обратите внимание на то, что код аутентичности сообщения не обеспечивает цифровую подпись, так как и отправитель, и получатель используют один и тот же общий ключ.

В табл. 8.2 представлена итоговая информация о степени конфиденциальности и аутентификации, обеспечиваемой различными подходами к решению этих вопросов.

Таблица 8.2. Использование кодов аутентичности сообщений

(а) $A \rightarrow B: \quad M \parallel C_K[M]$ • Обеспечивает аутентификацию - Только стороны **A** и **B** знают K
(б) $A \rightarrow B: \quad E_{K2}[M] \parallel C_{K1}[M]$ • Обеспечивает аутентификацию - Только стороны **A** и **B** знают K_1 • Обеспечивает конфиденциальность - Только стороны **A** и **B** знают K_2
(в) $A \rightarrow B: \quad E_{K2}[M] \parallel C_{K1}[E_{K2}[M]]$ • Обеспечивает аутентификацию - При использования K_1 • Обеспечивает конфиденциальность - При использования K_2

1.3. Функция хэширования.

Вариацией идеи использования кодов аутентичности сообщений является односторонняя функция хэширования. Как и в случае кодов аутентичности сообщений, функция хэширования получает на вход сообщение M произвольной длины, а на выход выдает хэш-код $H(M)$ фиксированного размера, иногда называемый профилем сообщения. Хэш-код является функцией всех битов сообщения и обеспечивает возможность контроля ошибок: изменение любого числа битов в сообщении выливается в изменение хэш-кода.

Способы использования хэш-кода для аутентификации сообщений показаны на рис. 8.5 и перечислены ниже.

1. Сообщение вместе с присоединенным к нему путем конкатенации хэш-кодом шифруется методами традиционного шифрования. С точки зрения общей схемы это аналогично стратегии контроля ошибок, показанной на рис. 8.2(а). Аргументация при этом тоже аналогична: так как только пользователям **A** и **B** становится известен секретный ключ, сообщение наверняка пришло от **A** и не могло быть изменено по пути следования. Хэш-код обеспечивает структуризацию или избыточность, требуемую для аутентификации. Поскольку шифрование выполняется по отношению ко всему сообщению вместе с добавленным хэш-кодом, при этом обеспечивается и конфиденциальность.
2. Шифруется только хэш-код средствами традиционного шифрования. Это позволяет снизить вычислительную нагрузку на систему, если речь идет о приложениях, не требующих конфиденциальности. Обратите внимание на то, что хэширование и шифрование в комбинации фактически дают код аутентичности сообщения (см. рис. 8.4(а)), т.е. $E_K[H(M)]$ представляет собой функцию сообщения M произвольной длины и секретного ключа K, которая дает на выходе значение фиксированного размера, защищенное от противника, не знающего секретного ключа.
3. Шифруется только хэш-код средствами шифрования с открытым ключом с использованием личного ключа отправителя. При этом обеспечивается не только аутентификация, как в предыдущем случае (б), но и цифровая подпись, так как только отправитель может произвести соответствующим образом зашифрованный хэш-код. Фактически в этом и заключается суть техники использования цифровой подписи.
4. Если требуется обеспечение не только конфиденциальности, но и цифровой подписи, можно зашифровать сообщение вместе с хэш-кодом, шифрованным открытым ключом. Для этого используются методы традиционного шифрования с секретным ключом.
5. В целях аутентификации сообщений можно использовать функцию хэширования без шифрования. В таком случае предполагается, что две участвующие в обмене данными стороны используют известное только им секретное значение S. Отправитель **A** вычисляет значение функции хэширования для результата конкатенации M и S и присоединяет полученное значение функции хэширования к M. Получателю **B** значение S известно, поэтому он может тоже вычислить значение функции хэширования, чтобы сравнить последнее с пришедшим вместе с сообщением. Ввиду того что секретное значение непосредственно не посылается, противник не может модифицировать перехваченное сообщение или генерировать ложное.
6. Конфиденциальность может быть обеспечена при некоторой модификации подхода описанного в пункте (д), если зашифровать сообщение вместе с добавленным к нему хэш-кодом.

Когда конфиденциальность не требуется, методы (б) и (в) оказываются предпочтительнее по сравнению с методами, в которых предполагается шифрование всего сообщения, поскольку требуют меньше вычислений. И, тем не менее, наблюдается постоянно возрастающий интерес к методам, которые позволяют избежать шифрования (рис. 8.5(д)). Это вызвано целым рядом причин.

- Программное обеспечение, выполняющее шифрование, работает довольно медленно. Даже если объем данных, которые шифруются при передаче сообщения, будет небольшим, поток исходящих и входящих сообщений в системе может оказаться очень интенсивным.
- Цены на аппаратные средства шифрования довольно высокие. Хотя и имеются недорогие микросхемы, реализующие алгоритмы DES, общая их стоимость может оказаться очень высокой, если задаться целью оснастить ими все узлы сети.
- Аппаратные средства шифрования оптимизируются для работы с большими объемами данных. При малых блоках данных значительная часть времени тратится непроизводительно на инициализацию/вызов.
- Алгоритмы шифрования могут быть защищены патентами. Некоторые алгоритмы шифрования, например алгоритм RSA шифрования с открытым ключом, запатентованы, и поэтому на их использование требуются лицензии, что тоже выливается в дополнительные расходы.
- Алгоритмы шифрования являются одним из вопросов экспортного государственного регулирования США.

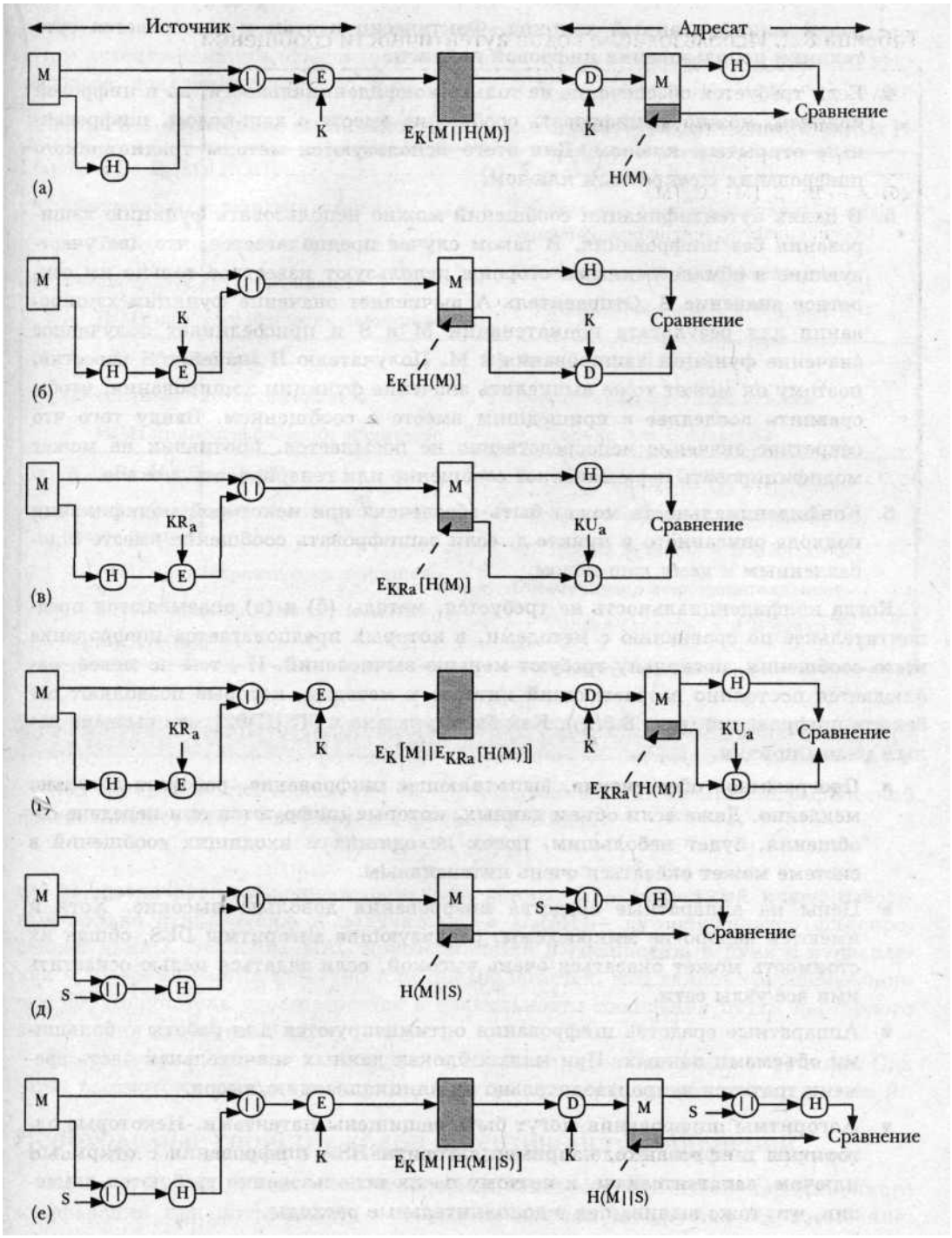


Рис. 8.5. Основные способы использования функции хэширования

В табл. 8.3 представлена итоговая информация о степени конфиденциальности и аутентификации, обеспечиваемой различными подходами к решению этих вопросов, показанными на рис. 8.5.

Таблица 8.3. Основные возможности использования функции хэширования H

(а) $A \to B:\ \ E_K[M \parallel H(M)]$ • Обеспечивает конфиденциальность - Только стороны **A** и **B** знают K • Обеспечивает аутентификацию - $H(M)$ криптографически защищено	**(г)** $A \to B:\ \ E_K[M \parallel E_{KRa}[H(M)]]$ • Обеспечивает аутентификацию и цифровую подпись • Обеспечивает конфиденциальность - Только стороны **A** и **B** знают K
(б) $A \to B:\ \ M \parallel E_K[H(M)]$ • Обеспечивает аутентификацию - $H(M)$ криптографически защищено	**(д)** $A \to B:\ \ M \parallel H(M \parallel S)$ • Обеспечивает аутентификацию - Только **A** и **B** знают S
(в) $A \to B:\ \ M \parallel E_{KRa}[H(M)]$ • Обеспечивает аутентификацию и цифровую подпись - $H(M)$ криптографически защищено - Только сторона **A** может создать $E_{KRa}[H(M)]$	**(е)** $A \to B:\ \ E_K[M \parallel H(M \parallel S)]$ • Обеспечивает аутентификацию - Только **A** и **B** знают S • Обеспечивает конфиденциальность - Только стороны **A** и **B** знают K