Программно-технические методы и средства защиты информации

В этой главе

- *Службы и механизмы защиты*
- *Обзор средств защиты информации в ИС*
- *Хеш-функции*
- *Цифровые подписи*
- *Механизмы неотказуемости (причастности)*
- *Режимы работы блочных алгоритмов шифрования*

<<< Этапы	Направления >>>	010				020				030				040				050			
		Защита объектов ИС				Защита процессов и программ				Защита каналов связи				ПЭМИН				Управление системой защиты			
	Основы >>>	База	Структура	Меры	Средства	База	Структура	Меры	Средства	База	Структура	Меры	Средства	База	Структура	Меры	Средства	База	Структура	Меры	Средства
		011	012	013	014	021	022	023	024	031	032	033	034	041	042	043	044	051	052	053	054
100	Определение информации, подлежащей защите	111	112	113	114	121	122	123	124	131	132	133	134	141	142	143	144	151	152	153	154
200	Выявление угроз и каналов утечки информации	211	212	213	214	221	222	223	224	231	232	233	234	241	242	243	244	251	252	253	254
300	Проведение оценки уязвимости и рисков	311	312	313	314	321	322	323	324	331	332	333	334	341	342	343	344	351	352	353	354
400	Определение требований к СЗИ	411	412	413	414	421	422	423	424	431	432	433	434	441	442	443	444	451	452	453	454
500	Осуществление выбора средств защиты	511	512	513	514	521	522	523	524	531	532	533	534	541	542	543	544	551	552	553	554
600	Внедрение и использование выбранных мер и средств	611	612	613	614	621	622	623	624	631	632	633	634	641	642	643	644	651	652	653	654
700	Контроль целостности и управление защитой	711	712	713	714	721	722	723	724	731	732	733	734	741	742	743	744	751	752	753	754

Программно-технические методы и средства (004) являются технической ОСНОВОЙ системы защиты информации. Применение таких средств осуществляется СТРУКТУРНЫМИ органами (002) в соответствии с принятой политикой информационной безопасности (003), описанной в нормативно- методических документах (001).

Программно-технические методы и средства (004) следует использовать В СЗИ по уже знакомым Вам НАПРАВЛЕНИЯМ:

- 010 Защита объектов корпоративных систем;
- 020 Защита процессов, процедур и программ обработки информации;
- 030 Защита каналов связи;
- 040 Подавление побочных электромагнитных излучений;
- 050 Управление системой защиты.

А для того, чтобы сформировать оптимальный комплекс (набор) программно-технических методов и средств защиты информации, необходимо пройти следующие ЭТАПЫ:

- 100 Определение информационных и технических ресурсов, подлежащих защите;
- 200 Выявление полного множества потенциально возможных угроз и каналов утечки информации;
- 300 Проведение оценки уязвимости и рисков информации при имеющемся множестве угроз и каналов утечки;
- 400 Определение требований к системе защиты;
- 500 Осуществление выбора средств защиты информации и их характеристик;
- 600 Внедрение и организация использования выбранных мер, способов и средств защиты;
- 700 Осуществление контроля целостности и управление системой защиты.

Совокупность защитных методов и средств включает в себя программные методы, аппаратные средства, защитные преобразования, а также организационные мероприятия (Рис. 9.1)

Сущность ***аппаратной или схемной защиты*** состоит в том, что в устройствах и технических средствах обработки информации предусматривается наличие специальных технических решений, обеспечивающих защиту и контроль информации, например экранирующие устройства, локализующие электромагнитные излучения или схемы проверки информации на четность, осуществляющей контроль за правильностью передачи информации между различными устройствами ИС.

Программные методы защиты — это совокупность алгоритмов и программ, обеспечивающих разграничение доступа и исключение несанкционированного использования информации.

Сущность ***методов защитных преобразований*** состоит в том, что информация, хранимая в системе и передаваемая по каналам связи, представляется в некотором коде, исключающем возможность ее непосредственного использования.

Организационные мероприятия по защите включают в себя совокупность действий по подбору и проверке персонала, участвующего в подготовке и эксплуатации программ и информации, строгое регламентирование процесса разработки и функционирования ИС.

Лишь комплексное использование различных защитных мероприятий может обеспечить надежную защиту, так как каждый прием или метод имеет свои слабые и сильные стороны.

Другой возможный вариант классификации методов защиты информации представлен на рис. 9.2.

Службы и механизмы защиты (004)

Служба защиты — совокупность механизмов, процедур и других управляющих воздействий, реализованных для сокращения риска, связанного с угрозой. Например, службы идентификации и аутентификации (опознания) помогают сократить риск угрозы неавторизованного пользователя. Некоторые службы обеспечивают защиту от угроз, в то время как другие службы обеспечивают обнаружение реализации угрозы. Примером последних могут быть службы регистрации или наблюдения.

Назовем некоторые службы защиты:

- ***идентификация и установление подлинности*** — служба безопасности, гарантирующая, что в ИС работают только авторизованные лица.
- ***управление доступом*** — служба безопасности, гарантирующая, что ресурсы ИС используются разрешенным способом.
- ***конфиденциальность данных и сообщений*** — служба безопасности, гарантирующая, что данные ИС, программное обеспечение и сообщения закрыты для неавторизованных лиц.
- ***целостность данных и сообщений*** — служба безопасности, гарантирующая, что данные ИС, программное обеспечение и сообщения не изменены неправомочными лицами.
- ***контроль участников взаимодействия*** — служба безопасности, гарантирующая, что объекты, участвующие во взаимодействии, не смогут отказаться от участия в

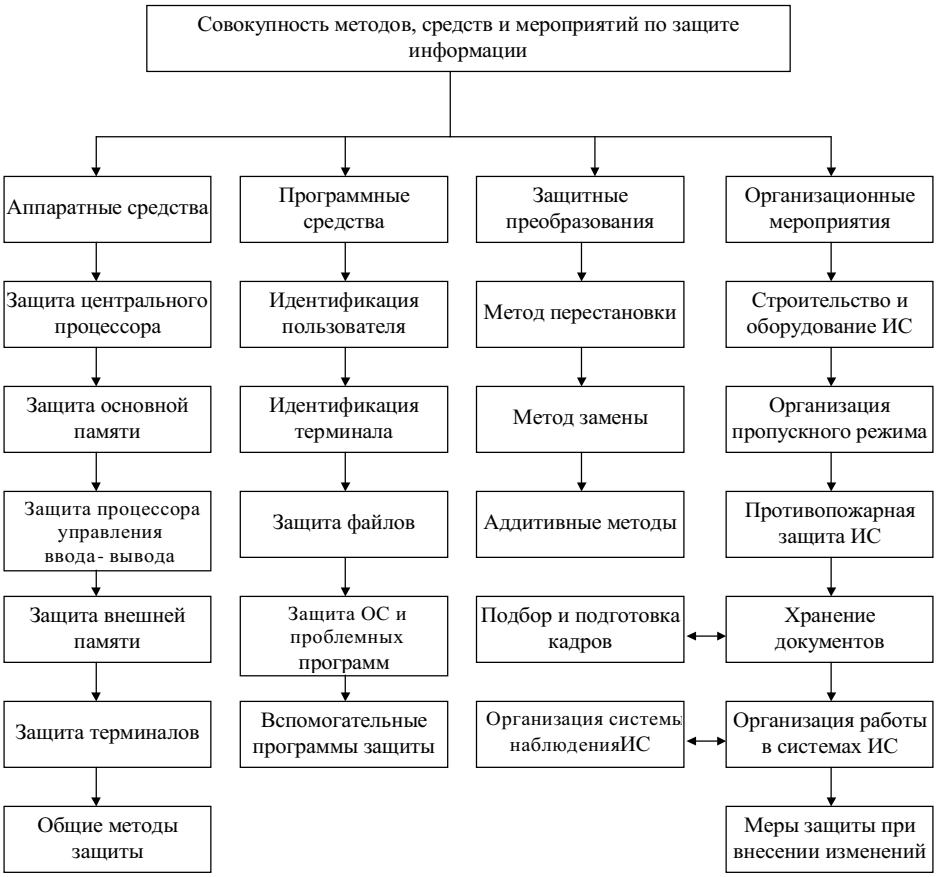


РИСУНОК 9.1. *Методы и средства защиты информации*

нем. В частности, отправитель не сможет отрицать посылку сообщения (контроль участников взаимодействия с подтверждением отправителя) или получатель не сможет отрицать получение сообщения (контроль участников взаимодействия с подтверждением получателя).

- ***регистрация и наблюдение*** — служба безопасности, с помощью которой может быть прослежено использование всех ресурсов ИС.

Приведенные в перечне службы защиты следует рассматривать как возможные, а не обязательные решения.

Методы идентификации и аутентификации пользователей (004)

Под *аутентификацией* пользователя (субъекта) понимается установление его подлинности.

Под *идентификацией* понимается определение тождественности пользователя или пользовательского процесса, необходимое для управления доступом. После идентификации обычно производится аутентификация.

Определение

Под авторизацией (санкционированием) подразумевается предоставление разрешения доступа к ресурсу системы.

При входе в систему пользователь должен предъявить идентифицирующую информацию, определяющую законность входа и права на доступ. Эта информация проверяется, определяются полномочия пользователя (аутентификация), и пользователю разрешается доступ к различным объектам системы (авторизация).

При наличии удаленных пользователей проблема аутентификации и контроля доступа носит критический характер для обеспечения безопасности всей сети.

Идентификация требует, чтобы пользователь был так или иначе известен ИС. Она обычно основана на назначении пользователю идентификатора пользователя. Однако ИС не может доверять заявленному идентификатору без подтверждения его подлинности. Установление подлинности возможно при наличии у

РИСУНОК 9.2. *Классификация методов защиты информации*

Аутентификация это установление подлинности пользователя...

пользователя уникальных особенностей и чем их больше, тем меньше риск подмены законного пользователя.

Требование, определяющее необходимость аутентификации, должно учитываться (явно или неявно) в большинстве политик безопасности ИС. Это требование может содержаться неявно в политике концептуального уровня, которая подчеркивает необходимость эффективного управления доступом к информации и ресурсам ИС, или может быть явно выражено в политике относительно ИС, в виде заявления, что все пользователи должны быть уникально идентифицированы и аутентифицированы.

В большинстве ИС используется механизм идентификации и аутентификации на основе схемы идентификатор пользователя/пароль. Аутентификация, которая полагается исключительно на пароли, часто не может обеспечить адекватную защиту. Пользователи часто выбирают пароли, легкие для запоминания и, следовательно, — для угадывания. С другой стороны, если пользователи вынуждены использовать пароли, сгенерированные из случайных символов, которые трудно угадать, то пользователям трудно их запомнить.

Программы проверки паролей — это программы, позволяющие определить, являются ли новые пароли легкими для угадывания и поэтому недопустимыми. Механизмы для использования только паролей, особенно те, которые передают по ИС пароль в открытом виде(в незашифрованной форме) уязвимы при наблюдении и перехвате. Это может стать серьезной пробле-

мой, если ИС имеет неконтролируемые связи с внешними сетями.

Учитывая уязвимость механизмов на основе паролей, целесообразно использовать более надежные механизмы, например системы аутентификации, основанные на смарт-картах и использовании биометрии.

Механизм, основанный на интеллектуальных картах, требует от пользователя владения смарт-картой и знания персонального кода идентификации (ПКИ-PIN) или пароль.

Смарт-карта реализует аутентификацию с помощью схемы запрос/ответ, использующей указанные выше параметры в реальном масштабе времени, что помогает предотвратить получение злоумышленником неавторизованного доступа путем воспроизведения сеанса регистрации пользователя. Эти устройства могут также шифровать сеанс аутентификации, предотвращая компрометацию информации аутентификации с помощью наблюдения и перехвата.

Механизмы блокировки для устройств ИС, автоматизированных рабочих мест или ПК на основе аутентификации пользователя могут быть полезны тем, кто должен часто оставлять рабочее место. Эти механизмы блокировки позволяют пользователям остаться зарегистрированными в ИС, не делая при этом свое рабочее место потенциально доступным злоумышленникам.

Модемы, которые обеспечивают пользователей доступом к ИС, требуют дополнительной защиты. Злоумышленник, получив доступ к модему, может получить доступ в ИС, угадав пароль пользователя.

Механизмы, обеспечивающие пользователя информацией об использовании его регистрационного имени, могут предупредить пользователя об использовании его имени необычным образом (например, многократные ошибки при регистрации). Эти механизмы включают уведомления о дате, времени и местоположении последнего успешного сеанса и числе предыдущих ошибок при регистрации.

Типы механизмов защиты, которые могли бы быть реализованы для обеспечения службы идентификации и аутентификации, приведены ниже:

- механизм, основанный на паролях,
- механизм, основанный на интеллектуальных картах,
- механизм, основанный на биометрии,
- генератор паролей,
- блокировка с помощью пароля,
- блокировка клавиатуры,
- блокировка ПК или автоматизированного рабочего места,
- завершение соединения после нескольких ошибок при регистрации,
- уведомление пользователя о “последней успешной регистрации” и “числе ошибок при регистрации”,
- механизм аутентификации пользователя в реальном масштабе времени,
- криптография с уникальными ключами для каждого пользователя.

Управление доступом (004)

Управление доступом может быть достигнуто при использовании дискреционного или мандатного управления доступом.

Дискреционное управление доступом — наиболее общий тип управления доступом, используемого в ИС. Основной принцип этого вида защиты состоит в том, что индивидуальный пользователь или программа, работающая от имени пользователя, имеет возможность явно определить типы доступа, которые могут осуществить другие пользователи (или программы, выполняемые от их имени) к информации, находящейся в ведении данного пользователя. Дискреционное управление доступом отличается от мандатной защиты тем, что оно реализует решения по управлению доступом, принятые пользователем.

Мандатное управление доступом реализуется на основе результатов сравнения уровня допуска пользователя и степени конфиденциальности информации.

Существуют механизмы управления доступом, поддерживающие степень детализации управления доступом на уровне следующих категорий:

- владелец информации,
- заданная группа пользователей,
- все другие авторизованные пользователи.

Это позволяет владельцу файла (или каталога) иметь права доступа, отличающиеся от прав всех других пользователей и определять особые права доступа для указанной группы людей или всех остальных пользователей.

В общем случае существуют следующие права доступа:

- доступ по чтению,
- доступ по записи,
- дополнительные права доступа (только модификацию или только добавление),
- доступ для выполнения всех операций.

Управление доступом пользователя может осуществляться на уровне каталогов или на уровне файлов. Управление доступом на уровне каталога приводит к тому, что права доступа для всех файлов в каталоге становятся одинаковыми. Например, пользователь, имеющий доступ по чтению к каталогу, может читать

(и,возможно, копировать) любой файл в этом каталоге. Права доступа к директории могут также обеспечить явный запрет доступа, который предотвращает любой доступ пользователя к файлам в каталоге.

В некоторых реализациях ИС можно управлять типами обращений к файлу помимо контроля за тем, кто может иметь доступ к файлу. Реализации могут предоставлять опцию управления доступом, которая позволяет владельцу помечать файл как разделяемый или заблокированный (монопольно используемый).

Разделяемые файлы позволяют осуществлять параллельный доступ к файлу нескольких пользователей одновременно. Блокированный файл будет разрешать доступ к себе только одному пользователю в данный момент. Если файл доступен только по чтению, назначение его разделяемым позволяет группе пользователей параллельно читать его.

Эти средства управления доступом можно использовать для ограничения допустимых типов взаимодействия между серверами в ИС. Большое количество операционных систем ИС могут ограничить тип трафика, посылаемого между серверами. Может не существовать никаких ограничений, что приведет к тому, что для всех пользователей будут доступны ресурсы всех серверов (в зависимости от прав доступа пользователей на каждом сервере). А могут и существовать некоторые ограничения, которые будут позволять только определенные типы трафика, например только сообщения электронной почты или более сильные ограничения, запрещающие трафик между определенными серверами.

Политика ИС должна определить, какими типами информации необходимо обмениваться между серверами. На передачу информации, не используемой совместно несколькими серверами, должны быть наложены ограничения.

Механизмы привилегий позволяют авторизованным пользователям игнорировать ограничения на доступ или, другими словами, легально обходить управление доступом, чтобы выполнить какую-либо функцию, получить доступ к файлу, и т.д. Механизм привилегий должен включать концепцию минимальных привилегий (принцип, согласно которому каждому субъекту в системе предоставляется наиболее ограниченное множество привилегий, необходимых для выполнения задачи).

Принцип минимальных привилегий должен применяться, например при выполнении функции резервного копирования. Пользователь, который авторизован выполнять функцию резервного копирования, должен иметь доступ по чтению ко всем файлам, чтобы копировать их на резервные носители информации. Однако пользователю нельзя предоставлять доступ по чтению ко всем файлам через механизм управления доступом.

Представим ***типы механизмов защиты для обеспечения службы управления доступом:***

- механизм управления доступом, использующий права доступа (определяющий права владельца, группы и всех остальных пользователей),
- механизм управления доступом, использующий списки управления доступом, профили пользователей и списки возможностей,
- управление доступом, использующее механизмы мандатного управления доступом,
- детальный механизм привилегий.

Конфиденциальность данных и сообщений (004)

Критичная информация может храниться в зашифрованном виде и если служба управления доступом будет обойдена, то информация останется недоступной для злоумышленника.

Очень трудно управлять неавторизованным доступом к трафику ИС.

Использование шифрования сокращает риск какого-либо перехвата и чтения проходящих транзитом сообщений, делая сообщения нечитаемыми для тех, кто сможет перехватить их. Только авторизованный пользователь, владеющий ключом, сможет расшифровать сообщение после его получения.

Политика безопасности должна явно указывать пользователям типы информации, которые считаются критичными настолько, что для них требуется применение шифрования, решение о его применении должно быть принято лицом в руководстве организации, ответственным за защиту критической информации. Если в политике не указывается, какая информация подлежит шифрованию, то владелец данных полностью отвечает за принятие решения.

Обеспечение конфиденциальности сообщений и данных (004)

Используемые механизмы безопасности:

- защита резервных копий на лентах, дискетах, и т.д.,
- физическая защита физической среды ИС и устройств,
- использование маршрутизаторов, которые обеспечивают фильтрацию для ограничения широковещательной передачи (блокировкой, маскированием содержания сообщения).

Обеспечение целостности данных и сообщений (004)

Служба целостности данных и сообщений помогает защитить данные и программное обеспечение на автоматизированных рабочих местах, файловых серверах и других компонентах ИС от неавторизованной модификации. Эта служба также помогает гарантировать, что сообщение не изменено, не удалено или не добавлено любым способом в течение передачи. Большинство современных методов защиты не могут предотвратить модификацию сообщения, но могут обнаружить модификацию сообщения (если сообщение не удалено полностью).

Можно также использовать электронные подписи для обнаружения модификации данных или сообщений. Электронная подпись может быть создана при помощи криптографии с открытыми или секретными ключами. При использовании системы с открытыми ключами документы в компьютерной системе подписываются с помощью электронной подписи путем применения секретного ключа отправителя документа. Полученная электронная подпись и документ могут быть затем сохранены или переданы. Подпись может быть проверена открытым ключом создателя документа.

Если подпись подтверждается должным образом, получатель может быть уверен в том, что документ был подписан с использованием секретного ключа его создателя и что сообщение не было изменено после подписания. Поскольку секретные ключи известны только их владельцам, это делает также возможным проверку личности отправителя сообщения третьим лицом. Поэтому электронная подпись обеспечивает две различных службы: контроль участников взаимодействия и целостность сообщения.

Типы механизмов защиты для обеспечения целостности данных и сообщений:

- коды аутентификации сообщения, используемые для программного обеспечения или файлов,
- использование электронной подписи, основанной на секретных ключах,
- использование электронной подписи, основанной на открытых ключах,
- детальный механизм привилегий,
- соответствующее назначение прав при управлении доступом (т.е. отсутствие ненужных разрешений на запись),
- программное обеспечение для обнаружения вирусов,
- бездисковые автоматизированные рабочие места (для предотвращения локального хранения программного обеспечения и файлов),
- автоматизированные рабочие места без накопителей для дискет или лент для предотвращения появления подозрительного программного обеспечения.

Регистрация и наблюдение (004)

Эта служба исполняет две функции.

Первая функция — обнаружение возникновения угрозы. Для всех обнаруженных случаев нарушения безопасности должна быть возможность проследить действия нарушителя во всех частях системы. Например, в случае вторжения злоумышленника в систему, журнал должен указывать, кто проводил сеанс с системой в это время, все критичные файлы, для которых имелись аварийно завершившиеся попытки доступа, все программы, которые запускались, и т.д.

Вторая функция — обеспечение системных и сетевых администраторов статистикой, которая показывает, что система и сеть в целом функционируют должным образом. Это может быть сделано при помощи механизма аудита, который использует файл журнала в качестве исходных данных и анализирует его с точки зрения корректного использования системы защиты.

Типы механизмов защиты для обеспечения регистрации и наблюдения:

- регистрация информации о сеансах пользователей (включая исходящую машину, модем, и т.д.),
- регистрация изменений прав пользователей для управления доступом,
- регистрация использования критичных файлов,
- регистрация модификаций, сделанных в критическом программном обеспечении,
- использование инструментов управления трафиком ИС,
- использование средств аудирования.

Регистрация действий пользователей (004)

Система защиты должна обеспечивать формирование контрольного журнала регистрации всех событий, связанных с доступом к ресурсам вычислительной системы. Такой журнал является непременным атрибутом вычислительных систем. Он способствует решению задач, среди которых:

- регулирование использования средств защиты в процессе работы системы;
- фиксация всех нарушений правил обращения к защищаемым данным;

- наблюдение за использованием реквизитов защиты (паролей, кодов и т.п.);
- обеспечение возврата к исходному состоянию системы при сбоях и других неисправностях;
- выполнение настройки элементов вычислительной системы, особенно библиотек программ и элементов баз данных в соответствии с частотой их использования.

Под сигнализацией в данном контексте понимаются следующие функции, реализуемые, как правило, программно:

- формирование и присваивание грифа секретности всем документам, выдаваемым на устройства отображения или печать;
- формирование и подача сигналов тревоги при обнаружении попыток НСД службе безопасности, администрации, операторам и пользователям; при этом для нарушителя можно имитировать нормальную работу с целью затягивания времени для принятия мер к обнаружению злоумышленника.

Единая регистрация при входе в ИС (004)

Существует громадное количество продуктов, обеспечивающих единую регистрацию (single sign-on, SSO), которые претендуют на решение данной проблемы. При работе многих из этих продуктов традиционно не уделялось внимания вопросам безопасности и интеграции с другими приложениями, но ситуация начинает меняться в лучшую сторону. Сейчас производители продуктов SSO предлагают вполне развитые программы и технологии, вполне отвечающие довольно жестким требованиям, связанным с условиями работы в крупных коммерческих предприятиях.

Главными задачами пользователей теперь становятся правильная формулировка своих требований и выбор такого продукта SSO, который наилучшим образом подходит для решения их задач.

Система SSO должна:

- иметь возможность поддержки других — помимо распознавания пользователей — средств безопасности, например обеспечения конфиденциальности и целостности сообщений;
- хорошо масштабироваться в сетевой среде всего предприятия;
- эффективно использовать сетевые ресурсы, такие как пропускная способность;
- обладать достаточным набором возможностей управления для успешного обслуживания всех членов разнообразного и рассредоточенного по предприятию пользовательского сообщества;
- использовать известные, повсеместно принятые стандартные алгоритмы шифрования, протоколы и интерфейсы прикладного программирования;
- в любой момент восстановить идентификатор или пароль пользователя;
- быстро и эффективно подключать к работе большое количество новых пользователей;
- поддерживать, помимо рабочих станций и серверов, такие элементы инфраструктуры, как маршрутизаторы, брандмауэры и системы управления сетью.

Контроль участников взаимодействия (004)

Служба контроля участников взаимодействия должна гарантировать, что субъекты взаимодействия не смогут отрицать участие во всем взаимодействии или какой-либо его части. Когда главной функцией ИС является электронная почта, эта служба безопасности становится особенно важной. Контроль участников взаимодействия с подтверждением отправителя дает получателю некоторую степень уверенности в том, что сообщение действительно прибыло от названного отправителя. Службу контроля участников взаимодействия можно обеспечить с помощью криптографических методов с использованием открытых ключей, реализующих электронную подпись.

Метод парольной защиты и его модификации (004)

Законность запроса пользователя определяется по паролю, представляющему собой, как правило, строку знаков. Метод паролей считается достаточно слабым, так как пароль может стать объектом хищения, перехвата, перебора, угадывания. Однако простота метода стимулирует поиск путей его усиления.

Для повышения эффективности парольной защиты рекомендуется:

- выбирать пароль длиной более 6 символов, избегая распространенных, легко угадываемых слов, имен, дат и т.п.;
- использовать специальные символы;
- пароли, хранящиеся на сервере, шифровать при помощи односторонней функции;
- файл паролей размещать в особо защищаемой области ЗУ ЭВМ, закрытой для чтения пользователями;
- границы между смежными паролями маскируются;
- комментарии файла паролей следует хранить отдельно от файла;
- периодически менять пароли;

- предусмотреть возможность насильственной смены паролей со стороны системы через определенный промежуток времени;
- использовать несколько пользовательских паролей: собственно пароль, персональный идентификатор, пароль для блокировки/разблокировки аппаратуры при кратковременном отсутствии и т.п.

В качестве более сложных парольных методов используется случайная выборка символов пароля и одноразовое использование паролей. В первом случае пользователю (устройству) выделяется достаточно длинный пароль, причем каждый раз для опознавания используется часть пароля, выбираемая случайно. При одноразовом использовании пароля пользователю выделяется не один, а большое количество паролей, каждый из которых используется по списку или по случайной выборке один раз.

В действительно распределенной среде, где пользователи имеют доступ к нескольким серверам, базам данных и даже обладают правами удаленной регистрации, защита настолько осложняется, что администратор мэйнфрейма все это может увидеть лишь в кошмарном сне.

Пример

"Чтобы обеспечить должную защиту на своем мэйнфрейме, нам потребовались годы проб и ошибок – сказал Кен Катлер (Ken Cutler), вице-президент Института защиты информации (Information Security Institute), который является филиалом Института подготовки администраторов информационных систем – MIS Training Institute (Фреймингхем, шт.Массачусетс). – Теперь мы пытаемся проделать все это снова, но в более трудных условиях, имея дело с возросшей сложностью и людьми, которые раньше не были связаны с системой".

Карл Аллен, научный сотрудник, занимающийся вопросами защиты в Novel (Прово, шт.Юта), считает, что меры по обеспечению безопасности можно разделить на несколько категорий, включая ограничение доступа к рабочим станциям, усовершенствование систем запрос/ответ, инструменты наблюдения и администрирования, системы шифрования и, наконец (самая новая область), единый пароль регистрации в сетях различных производителей. "Покупатели хотят, чтобы эти средства были более тесно интегрированы с NetWare", – добавил Аллен.

(Rachel Parker "Honey, did you lock the LAN?")

Подсистема управления ключами (004)

Подсистема управления СЗИ предназначена для управления ключами подсистемы криптографической защиты, а также контроля и диагностирования программно-аппаратных средств и обеспечения взаимодействия всех подсистем СЗИ.

Под управлением криптографическими ключами подразумеваются все действия, связанные с генерацией, распределением, вводом в действие, сменой, хранением, учетом и уничтожением ключей.

К функциям подсистемы управления ключами шифрования относятся:

- генерация, тестирование, учет и распределение ключей;
- контроль за хранением и уничтожением ключей;
- контроль за вводом в действие и сменой ключей;
- ведение базы данных открытых ключей (БД ОК) на центре распределения ключей (ЦРК);
- рассылка БД ОК пользователям;
- контроль за вводом в действие и сменой ключей цифровой подписи;
- контроль и диагностирование программно-аппаратных средств защиты.

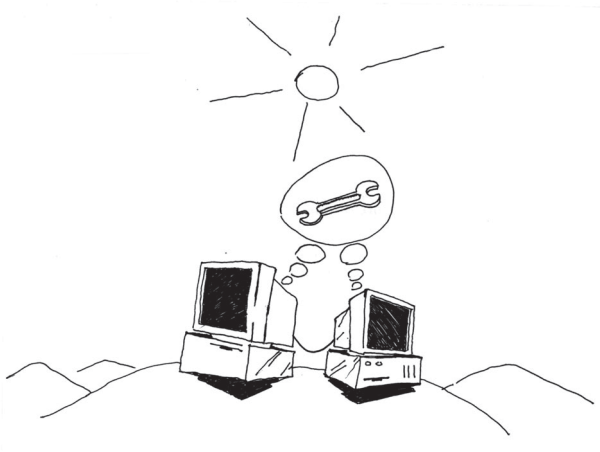
Подсистема состоит из центра распределения ключей и программно-аппаратных средств, интегрированных в рабочие станции пользователей.

В СЗИ ИС управление ключами возлагается на центр распределения ключей. ***ЦРК осуществляет:***

- централизованную генерацию симметричных шифрключей, их распределение и контроль за дальнейшим использованием;
- ведение и рассылку базы данных открытых ключей;
- контроль за использованием несимметричных ключей;
- ведение архивов открытых ключей цифровой подписи;
- участие в предварительной проверке спорных ситуаций, возникающих при использовании цифровой подписи;
- разработку мероприятий на случай компрометации ключей;
- гарантированное стирание ключевых данных на носителях по истечении срока действия ключей.

В качестве ключевой схемы целесообразно выбрать двухуровневую (главный ключ, формируемый на ЦРК, плюс сеансовый ключ, формируемый пользователем).

Ключи цифровой подписи рекомендуется формировать самим пользователям, чтобы не создавать проблему доверия к ЦРК. Последний осуществляет управление открытыми ключами цифровой подписи. При этом формируется база данных открытых ключей, которая рассылается всем пользователям ИС, применяющим цифровые подписи. ЦРК следит за обновлением базы, контролирует ввод в действие и срок действия ключей цифровой подписи, разрабатывает мероприятия на случай компрометации ключей.



Подсистема управления ключами....

Биометрические средства аутентификации и контроля доступа (004)

Расширением парольного метода является опознавание пользователя по сугубо индивидуальным характеристикам. Эти методы, как правило, требуют специального и достаточно сложного оборудования. ***Известны такие методы:***

а) персональные: отпечатки пальцев; строение лица;

б) квазистатические: геометрия руки; особенность глаз; отпечатки ладоней; рисунок кровеносных сосудов;

в) квазидинамические: пульс; баллистокардиография; энцефалография;

г) динамические: голос; почерк; стиль печатания.

Широкое распространение получили средства опознавания атрибутного типа, изготовляемые в виде карточек. Карточка является носителем идентификационной информации, нанесенной механическим, оптическим или магнитным способом.

На смену магнитным карточкам приходят более устойчивые к подделке “интеллектуальные карточки” (ИК) (smartcard), содержащие электронные компоненты (микропроцессор, энергонезависимая память). Существует международный стандарт на ИК -ISO 7816. Устройство ИК позволяет многократную запись/чтение содержимого памяти. ***Карточку можно использовать для хранения:***

- идентификационной информации;
- ключей шифрования и использования в качестве криптопроцессора;
- любой конфиденциальной информации.

Некоторые карточки обеспечивают режим “самоблокировки” при попытке НСД.

Метод автоматической генерации обратного вызова (004)

Сущность метода в том, что центральная ЭВМ, получив вызов от пользователя, в котором содержится персональный код типа пароля, прерывает связь с пользователем и проверяет этот код. При положительном результате проверки программа вызывает пользователя по телефону и подключает к ЭВМ.

Широкое применение получили модемы с обратным вызовом. В этом случае после установления соединения отвечающий модем разрушает соединение и набирает хранящийся в его памяти номер вызывающего модема. Этот вид защиты используется только при работе по коммутируемым каналам связи.

Метод перекрестного опознавания (004)

Процедура опознавания повторяется периодически в процессе работы пользователя, причем моменты повторения процедуры выбираются случайно. При этом каждый раз могут быть использованы различные методы опознавания.

Процедура аутентификации может быть обращенной: в данном случае объект сообщает сведения о себе субъекту, например для уверенности пользователя, что в идентифицированную систему можно безопасно вводить конфиденциальную информацию. В качестве приемника данных можно рассматривать центральную ЭВМ, управляющую процедурой аутентификации и контроля доступа при входе в систему, и корреспондентов, которым посылается информация.

Обращенная процедура предотвращает угрозу типа “маскарад”: если защищаемая информация обрабатывается в территориально распределенной сети, то злоумышленник со своего терминала может попытаться перехватить запрос пользователя и начать работу с ним, имитируя работу ИС и получая таким образом от пользователя всю посылаемую информацию.

Интересно

Имитация работы системы может происходить в виде обычных запросов, набираемых злоумышленником или с применением специальных программ, формирующих стандартную картинку запроса системы (“ложный экран”). В результате злоумышленных действий осуществляется переадресация передачи информации. Существует несколько методов аутентификации приемника данных.

Проверка адреса корреспондента (004)

При несанкционированном изменении адреса корреспондента, хранящегося в регистре или зоне ЗУ, передача

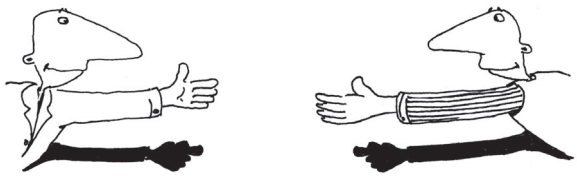
идет по ложному адресу. Рекомендуется в процессе передачи периодически проверять адрес корреспондента, сравнивая его с эталоном, хранящимся в защищенной зоне ЗУ. При несовпадении сравниваемых адресов передача данных блокируется и вырабатывается соответствующий системный сигнал.

Проверка обратного кода (004)

Процедура защиты заключается в том, что у корреспондента периодически запрашивается идентифицирующая информация, называемая обратным кодом. Последний сравнивается с эталоном; при несовпадении передача блокируется. В этом случае обнаруживается не только злоумышленная переадресация данных, но и несанкционированное использование терминала зарегистрированного пользователя.

Схема рукопожатия (004)

Наиболее эффективным решением проблемы аутентификации системы и ее элементов считается реализация "схемы рукопожатия". При ее применении заранее выбирается нетривиальное преобразование A(x,k), где k — ключ, зависящий от времени, а x — аргумент. Предполагается, что преобразование известно только пользователю и системе. При запросе на работу пользователь вводит аргумент x. Система и пользователь вычисляют преобразование A(x,k), затем система посылает свой результат пользователю для сравнения. При совпадении обоих преобразований аутентификация считается положительной.



Наиболее эффективным решением проблемы аутентификации считается "схема рукопожатия"...

Перспективным предполагается использование для аутентификации пользователей применение электронных ключей. Это компактные устройства, которые подключаются к одному из внешних разъемов компьютера, "прозрачные" для периферийных устройств. Наиболее совершенные ключи программируются уникальным образом фирмой-изготовителем или самим пользователем. С их помощью можно не только хранить и возвращать информацию по запросу, но и аппаратно реализовывать некоторое ее преобразование.

Внешне электронный ключ представляет собой схему, собранную по SMD-технологии (Surfase Mounted Device), чаще всего на основе специальной "заказной" микросхемы, с разъемами для подключения к компьютеру и внешним устройствам, помещенную в пластмассовый непрозрачный корпус.

По сложности для эмулирования ключи можно разделить на три группы в зависимости от строения входных и выходных данных.

1. Простейшие — работают по принципу "есть ключ — нет ключа";

2. Стандартные — работают по принципу внешнего запоминающего устройства, доступного для чтения заранее записанной информации;

3. Сложные — устроены по принципу аппаратно реализованной математической функции;

Контроль доступа пользователей к ресурсам ИС (004)

Контроль доступа — это предотвращение несанкционированного использование ресурса системы, включая его защиту от несанкционированного использования. Контроль доступа тесно связан с аутентификацией пользователей (устройств), поскольку именно в процессе опознавания предъявляются полномочия на доступ к различным объектам системы (массивам, данным, техническим средствам).

В соответствии с критериями оценки защищенных вычислительных систем, принятыми в США и России, должна предусматриваться реализация избирательного и/или мандатного (полномочного) контроля доступа.

Сущность избирательного контроля доступа (ИКД) состоит в следующем (на примере СУБД):

ИКД ограничивает возможность пользователя выполнять определенные операции или получать доступ к определенному объекту на основе его привилегий. Так, примерами объектов в СУБД являются таблицы, последовательности и представления. Поскольку санкционированные пользователи могут передавать свои привилегии другим по своему усмотрению, этот тип контроля доступа называют избирательным.

Сущность мандатного контроля доступа (МКД) состоит в следующем:

МКД опосредует доступ к информации на основе ее секретности и допуска пользователя, пытающегося получить доступ к ней. Если ИКД основан (и может быть скомпрометирован) по усмотрению пользователя, то МКД гарантирует, что секретная информация не будет распространяться среди пользователей, не имеющих санкции на ее получение. Система, реализующая МКД, называется системой с многоуровневой защитой (МУЗ).

Системы с МУЗ хранят метки для каждого объекта системы. Эта метка представляет степень секретности информации, хранимой в объекте. Системы с МУЗ так-

же хранят допуск для каждого пользователя. Этот допуск определяет диапазон меток, к которым имеет доступ пользователь. Система гарантирует, что пользователи не получат доступа к информации за рамками их допуска или каким-либо иным способом.

Мандатный контроль доступа реализуется в вычислительных системах, в которых обрабатывается информация различных грифов секретности.

Методы предотвращения повторного использования объектов (004)

Подсистема защиты от НСД СЗИ должна предотвращать попытки незаконного получения критичной информации, остатки которой могли сохраниться в некоторых объектах памяти (блоки файлов, буферы), ранее использованных другим пользователем. Это достигается путем стирания информации или отказа в чтении "свободного" блока, пока пользователь не заполнит его информацией. Наиболее опасно в данном контексте чтение ключей, паролей и другой аутентифицирующей информации.

Уничтожение может быть надежно осуществлено двухтрехкратной записью в объектах памяти случайной комбинации из нулей и единиц после каждого запроса.

Обзор средств защиты информации в ИС (004)

Невысокая скорость распространения систем компьютерной безопасности объясняется тем, что при широком внедрении корпоративных сетей лишь малое число специализированных фирм может предоставить квалифицированные услуги в области системной интеграции средств безопасности. Применение же отдельных средств не дает эффекта, но "съедает" бюджетные средства.

К сожалению, в наши дни типично скорей интуитивное проявление интереса, например, к способам обезопасить свои данные, а не целенаправленная работа по анализу уровня безопасности своих информационных систем. Это результат того, что лишь крупные корпоративные заказчики могут проводить дорогостоящие проекты по защите информации и содержать штат сотрудников, занимающихся развитием и поддержкой защитных систем. Поэтому очень важно понимание заказчиком своих потребностей по защите информации, проблем организации работ и своих финансовых возможностей.

Предлагается выделить следующие ***группы средств защиты информации***:

- средства защиты от НСД;
- системы анализа и моделирования информационных потоков (CASE-системы);
- системы мониторинга сетей;
- анализаторы протоколов;
- антивирусные средства;
- межсетевые экраны;
- криптографические средства;
- системы резервного копирования;
- системы бесперебойного питания;
- системы аутентификации;
- средства предотвращения взлома корпусов и хищения оборудования;
- средства контроля доступа в помещения;
- инструментальные средства анализа системы защиты.

В результате интенсивного развития информационных технологий, увеличения размеров информационных сетей и огромного количества прикладного и системного ПО стала очевидной необходимость использования таких средств, как системы мониторинга сетей и системы анализа информационных потоков. Причем если для небольших сетей можно ограничиться лишь системой мониторинга, то для крупных корпоративных сетей необходимы оба класса продуктов.

Использование систем анализа и моделирования информационных потоков заслуживает отдельного обсуждения. Эти сравнительно новые на нашем рынке продукты, предлагаемые по вполне приемлемым ценам, действительно необходимы в процессе создания глобальной системы анализа крупных сетей, но для их грамотного использования необходим обученный специалист, что не всегда возможно.

Служба информационной безопасности использует следующие средства:

- средства защиты от НСД;
- анализаторы протоколов;
- инструментальные средства анализа системы защиты;
- межсетевые экраны;
- криптографические средства;
- системы аутентификации;
- средства предотвращения взлома корпусов и хищения оборудования;
- средства контроля доступа в помещения.

Применение этих средств в разной мере необходимо для обеспечения безопасности информации в автоматизированных системах. Рассмотрим некоторые из них.

Средства защиты от НСД (004)

Средства этого направления широко представлены на рынке. В основном они представляют собой программно-аппаратные комплексы с применением личного идентификатора (электронный идентификатор семейства Touch Memory (iButton), микропроцессорная карта и т.д.). Продукты этого класса позволяют разграничивать доступ к информационным ресурсам вычислительной техники, вести аудит сеансов работы, администрировать используемые программные средства. Кроме этого, некоторые из них имеют встроенные антивирусные функции и средства криптографической защиты информации. При сетевом использовании защищаемых рабочих мест имеется возможность удаленного администрирования каждого из них и получение полной статистики по попыткам доступа к компьютеру и сеансах работы.

Анализаторы протоколов (004)

В процессе управления и решения задач безопасности сетей часто возникает вопрос о сборе информации, декодировании и статистическом анализе информации с помощью сетевых протоколов разных уровней. В случае администрирования небольших корпоративных систем, потребности администратора безопасности вполне могут удовлетворить портативные анализаторы серии Expert Sniffer Analyzer (ESA), известные также и под названием Turbo Sniffer Analyzer. Выпускаемые в настоящее время версии продуктов обеспечивают полный анализ, интерпретацию протоколов, а также мониторинг подключенного к анализатору сегмента сети.

Пример

Обеспечение безопасности распределенных информационных систем, разработанных в рамках идеологии "клиент-сервер", ставит перед администратором безопасности задачу анализа пакетов не только на уровнях модели OSI ISO, но и на уровне специфики пакетов "клиент-сервер". Например, в случае построения распределенной информационной системы на базе СУБД ORACLE 7 возникает проблема анализа пакетов протокола SQL*Net v.2. Эта проблема может быть успешно решена при помощи модуля Sniffer Network Analyzer Database Module (for Oracle7).

Следует отметить, что программным анализаторам протоколов ИС при всем удобстве работы с ними, свойственен существенный недостаток, связанный с необходимостью использования выделенной рабочей станции для выполнения задач по анализу сетевого трафика. Это решение не всегда приемлемо по причине жесткой привязки анализатора к топологии сети.

Практика показывает, что корпоративная сеть предприятия представляет собой достаточно живой организм, и трудно заранее определить тот участок сети, который нуждается в повышенном уровне контроля со стороны администратора безопасности. Необходимость установки стационарных анализаторов в конкретных точках корпоративной сети определяется в соответствии с политикой безопасности, принятой на данном предприятии.

В большинстве практически значимых случаев целесообразно производить детальный анализ трафика в точках сопряжения с внешними каналами связи, а также в точках подключения ответственных сегментов к серверам (server farm). В любом случае следует учитывать, что эксплуатация стационарного анализатора трафика представляет собой достаточно сложный процесс, требующий участия в нем квалифицированных специалистов.

Администратор безопасности для качественного выполнения своих функций, связанных с практическим воплощением правил политики безопасности, испытывает необходимость в наличии мобильного и легкого в эксплуатации ***анализатора состояния сети. Только наличие такого прибора позволяет администратору безопасности быстро и квалифицированно идентифицировать причину и источник нарушения принятых на предприятии правил безопасности.***

Применение таких приборов позволяет:

- производить анализ и контроль глобальных сетей, базирующихся на протоколах типа ETHERNET и TOKEN RING;
- выявлять наиболее активных отправителей (получателей) данных, а также отправителей широковещательных пакетов;
- анализировать ошибочные пакеты по типу ошибки и источнику пакета;
- собирать информацию о коллизиях (а для протокола TOKEN RING — о времени обращения маркера и неполадках при реконфигурации кольца);
- осуществлять испытания сети на пропускную способность путем моделирования повышенного уровня трафика методом генерации пакетов;
- выявлять наиболее уязвимые участки сети с точки зрения возможности организации атак на доступность ("забрасывание" маршрутизаторов сетевыми пакетами, переполнение буферных областей межсетевых экранов, не обладающих механизмом кэширования, инициация перезапуска мостов и прочие популярные атаки на доступность ресурсов ИС);
- осуществлять выборочное тестирование мостов и маршрутизаторов;
- выполнять детальный анализ топологии сети, включая хосты, маршрутизаторы, удаленные сегменты, а

также выявлять дубли адресов, которые могут образовываться вследствие наличия паразитных подключений злоумышленников;

- производить исчерпывающее исследование сетей NETWARE, включая определение типа используемого кадра, определение списка доступных и ближайших серверов, а также помогают при установке и конфигурации клиентской части;
- собирать статистические данные, которые включают в себя информацию об источниках и приемниках пакетов, частоте обращений к конкретным файлам и иные данные, необходимые при осуществлении мониторинга безопасности корпоративной сети.

Помимо анализа протоколов, приборы такого класса обладают функциональностью кабельного тестера. С помощью дополнительного устройства, подключаемого к дальнему концу кабеля, определяются такие характеристики кабельной системы, как длина кабеля, ошибки в разводке (наличие расщепленных и скрещенных пар), расстояние до неисправности (величина перекрестной наводки), наличие обрыва или короткого замыкания, а также многие другие свойства кабельной системы.

Инструментальные средства тестирования системы защиты (004)

Систему защиты корпоративной сети можно считать достаточно надежной только при условии проведения постоянного тестирования.

В идеале администратор безопасности должен собирать информацию о возможных атаках, систематизировать ее и периодически осуществлять проверки системы защиты путем моделирования возможных атак. Очевидно, что выполнение этой задачи в полном объеме требует привлечения огромных материальных средств. Но можно обойтись значительно меньшими затратами, если прибегнуть к услугам фирм, специализирующихся на производстве устройств проверки надежности систем защиты. К таким фирмам относится, например, компания INTERNET SECURITY SYSTEM, обладающая правами на программу INTERNET SCANNER, а также программу SATAN (Security Administrator Tool for Analyzing Networks).

В настоящее время наиболее развитым продуктом тестирования уровня защиты корпоративных сетей является система Internet Scanner SAFEsuite, разработанная фирмой ISS. Этот продукт предоставляет администратору безопасности возможность всесторонней проверки уровня реализации политики безопасности.

Пример

Остановимся на описании возможностей некоторых модулей, входящих в состав системы Internet Scanner SAFEsuite.

Модуль Web Security Scanner осуществляет *поиск уязвимых мест* в настройках WEB-серверов и выявляет подозрительные CGI-скрипты.

Firewall Scanner осуществляет всестороннее *тестирование межсетевых экранов* и приложений, обращения к которым осуществляется через межсетевые экраны. В настоящее время Internet Scanner SAFEsuite обладает возможностью тестирования двух десятков типов межсетевых экранов и обладает исчерпывающим описанием примерно 140 типов возможных атак непосредственно на межсетевые экраны. Администратор безопасности обладает возможностью коррекции базы данных модуля Firewall Scanner путем введения описаний новых возможных атак.

Модуль System Security Scanner осуществляет контроль систем защиты персональных компьютеров, анализирует оптимальность параметров настройки операционных систем, контролирует порядок реализации прав доступа к файлам в соответствии с принятой на предприятии политикой безопасности, производит поиск программ типа “троянский конь”. Этот модуль способен произвести детальный анализ безопасности конкретных состояний таких операционных систем, как NT, UNIX, WINDOWS 95, а также программного обеспечения шлюзов и маршрутизаторов.

Межсетевые экраны (004)

Межсетевые экраны (FireWall-система или Брандмауер) — это программные продукты, используемые для защиты от несанкционированных действий со стороны внешней сети (INTERNET — INTRANET продукты) и для разделения сегментов корпоративной сети (ENTERPRISE продукты). Так, система FIREWALL/PLUS-LE является программным продуктом, относящимся к классу межсетевых экранов, и предназначена для обеспечения безопасности компьютерных сетей.

Система работает под управлением ОС WINDOWS NT, причем следует отметить, что она была разработана именно для этой ОС, а не перенесена на нее. В качестве интерфейса администрирования используется система WINDOWS. Управление системой осуществляется как локально, так и со стороны защищаемой сети. Система поддерживает модель использования виртуальных частных сетей.

Работа системы в качестве средства защиты осуществляется на трех уровнях:

- фильтрация пакетов;
- шлюзование уровня приложения;
- шлюзование низкого уровня.

В случае применения механизмов фильтрации пакетов пользователю предоставляется возможность использовать уже имеющиеся и создавать нестандартные фильтры, используя современный графический интерфейс с интуитивно понятной visual-системой, не прибегая к программированию на алгоритмических языках.

Шлюзование уровня приложения позволяет следить за сеансом работы программы и вести его аудит. Эта возможность широко используется для наложения ограничений на трафик и конкретное приложение

Шлюзование низкого уровня позволяет защищать сетевые ресурсы, связанные с внешним TCP/IP портом. Это средство контролирует допустимость связи по протоколам TCP/IP и UDP, не идентифицируя конкретное приложение.

Вся информация о сеансах работы протоколируется. Журнал содержит данные о предоставляемых INTERNET-услугах, временных метках событий, источниках пакетов, объемах передачи, объемах приема, продолжительности подключения.

Хеш-функции (034)

Общие сведения и классификация хеш функций (034)

Криптографические хеш-функции играют фундаментальную роль в современной криптографии. Особенно широко они используются при обеспечении целостности данных и аутентификации сообщений. Аутентичность сообщения можно обеспечить различными способами, не прибегая к его шифрованию. Такой подход пригоден во многих случаях, когда целостность и аутентичность данных играет исключительно важную роль, а конфиденциальность не требуется, например, при реализации финансовых операций и распределении открытых ключей между объектами.

Широко распространены следующие методы обеспечения подлинности сообщения:

- добавление к сообщению кода подлинности сообщения (код аутентификации сообщения) (message authentification code, MAC-код) или зашифрованной контрольной суммы;
- введение цифровых подписей.

Далее будем считать, что подтверждение подлинности означает в первую очередь способность получателя проверить, что сообщение не изменено третьим лицом, не является повтором ранее переданного сообщения или фальшивым сообщением, созданным третьим лицом.

Необходимо различать функцию ***аутентификации пользователя*** и функцию ***аутентификации сообщения.*** Вхождение пользователя в вычислительную систему естественно рассматривать как начало сеанса работы с терминалом и сопроводить его выполнением процедуры подтверждения подлинности пользователя. Задача обеспечения подтверждения подлинности пользователя решается системами управления доступа и реализуется через услуги и механизмы управления доступом.

В сетях связи более важным является сообщение: субъекты сети обмениваются сообщениями, и аутентификация источника и содержимого сообщения должна быть выполнена при получении каждого нового сообщения. Соответствующая функция защиты называется ***аутентификация источника сообщений*** (см. ISO 7498-2).

Функция, в общем случае, должна подтверждать следующие факты:

- сообщение исходит от санкционированного отправителя;
- содержание сообщения при передаче не изменилось;
- сообщение доставлено по адресу;
- аналогичное сообщение ранее не поступало;
- порядок получения сообщений соответствует порядку отправления.

В случае конфликтной ситуации третье лицо (посредник) должно удостоверить, что действительно сообщение послано одним санкционированным субъектом сети другому т.е. реализуется и функция неотказуемости (причастности). Одним из составных элементов механизмов безопасности реализующих функции целостности, аутентификации и причастности как раз и являются хеш-функции.

Хеш-функция берёт на вход сообщение и порождает на выходе некоторый образ этого сообщения, который называется хеш-кодом, хеш-результатом, хеш-значением или просто хеш. Или более точно, хеш-функция h отображает двоичную строку произвольной конечной длины m в двоичную строку фиксиро-

Хеш-функция обеспечивает целостность данных...

ванной длины, скажем n. В криптографии используется именно эта идея, т.е. когда хеш-код выступает в роли компактного представления (образа) некоторой входной строки, по которому можно точно идентифицировать исходное сообщение.

Хеш-функции можно разделить на два класса: бесключевые хеш-функции, т.е. хеш-функции на вход которых подается только сообщение и ключевые хеш функции, т.е. хеш-функции на вход которых подается сообщение и секретный ключ.

Для дальнейших рассуждений приведем следующее определение хеш-функции.

Определение

Хеш-функция, в самом общем смысле, есть функция h(x) которая как минимум обладает следующими двумя свойствами:

- сжатие – т.е. функция h отображает входную строку x конечной произвольной длины в выходную строку $y = h(x)$ фиксированной длины n;
- легкость вычисления – при известной h и входной строке x легко вычислить $h(x)$.

К бесключевым хеш-функциям относятся ***коды обнаружения изменений сообщения*** (MDC-код, modification detection code), также известные как коды обнаружения манипуляций над сообщениями или коды целостности сообщений. MDC-коды предназначены для формирования сжатого образа или хеш-кода сообщения, который удовлетворяет специальным свойствам. В конечном итоге MDC-коды обеспечивают, совместно с другими механизмами, целостность данных. В свою очередь MDC-коды могут быть разбиты на ***односторонние хеш-функции***, для которых сложно найти входное значение по известному хеш-коду и ***стойкие к столкновениям хеш-функции***, для которых сложно найти два входных значения имеющих один и тот же хеш-код. Бесключевые хеш-функции являются одним из составных элементов цифровых подписей.

К ключевым хеш-функциям относятся MAC-коды. Они предназначены для обеспечения целостности данных и аутентификации сообщений без использования каких-либо других механизмов и позволяют обеспечить аутентификацию сообщения на основе использования методов симметричной криптографии. Алгоритмы формирования MAC-кодов рассматриваются как хеш-функции с двумя входными параметрами, а именно сообщением и секретным ключом. На выходе такого алгоритма формируется двоичная строка фиксированной длины. При этом на практике невозможно сформировать точно такую же строку без знания ключа. MAC-коды могут использоваться как для обеспечения целостности данных, так и аутентификации источника данных.

В криптографических приложениях общепринятым является то, что алгоритмы хеш-функций являются открытыми. Таким образом, в случае использования MDC-кода по данному входному сообщению хеш-код может вычислить любой субъект, а при использовании MAC-кода вычислить хеш-код по данному входному сообщению может только субъект, обладающий секретным ключом.

Стандарты кодов аутентификации сообщений (034)

Первыми стандартами на MAC-коды являются американские национальные стандарты ANSI X9.9 и ANSI X9.19, опубликованные соответственно в 1982 и 1986 годах.

В 1986 году появился международный стандарт ISO 8730, который является по сути международным эквивалентом ANSI X9.9 и определяет общие требования для таких механизмов. В дополнение к ISO 8730, были разработаны стандарты ISO 8731-1:1987 и ISO 8731-2:1987. Стандарты ANSI X9.9, ANSI X9.19, ISO 8730, ISO 8731-1:1987 определяют алгоритмы формирования MAC-кодов на основе использования алгоритма блочного шифрования DES в CBC режиме. Такой код подлинности известен в литературе как CBC-MAC. Стандарт ISO 8731-2:1987 стандартизирует алгоритм аутентификации сообщения (Message Authentificator Algorithm, MAA). Перечисленные стандарты ориентированы на применение стандартизируемых методов формирования MAC-кодов в банковских технологиях. На основе банковских стандартов в 1989 году ISO разрабатывает стандарт MAC-кода общего назначения — ISO/IEC 9797. Этот стандарт также использует блочный шифр в CBC режиме, то есть он определяет CBC-MAC. В 1994 году стандарт ISO/IEC 9797 был доработан и обновлен.

В 1997 начался пересмотр международного ISO стандарта MAC-кода. Существующий стандарт 1994 года был заменён на стандарт ISO/IEC 9797-1, содержащий расширенное множество CBC-MAC механизмов. Другая часть стандарта, ISO/IEC 9797-2, также пересмотренная и улучшенная, содержит ряд механизмов формирования MAC-кодов на базе хэш-функции, включая метод HMAC.

MAC-коды (034)

Алгоритм формирования кода аутентификации сообщения есть семейство функций h_k, где k — секретный ключ, обладающих следующими свойствами:

1. Простота вычисления — для для известной функции hk, заданного значения k и входного значения x, лекго вычислить $h_k(x)$. Полученный результат называется MAC-код.

2. Сжатие — h_k отображает входное значение x — конечную двоичную строку произвольной длины в выходное значение $h_k(x)$ — двоичную чтроку фиксированной длины n.

3. Стойкость к вычислению — по известным парам "текст — MAC-код" (x_i, $h_k(x_i)$) вычислительно невозможно вычислить любую другую пару "текст-MAC-код" (x', $h_k(x')$) для любого нового входного значения $x' \neq x_i$.

Если последнее свойство не выполняется, то возможна подделка MAC-кода. В то время как стойкость к вычислению подразумевает и наличие свойства не восстанавливаемости ключа (т.е. вычислительно невозможно восстановить ключ k, по известной одной или более парам "текст-MAC-код" (x_i, $h_k(x_i)$), полученных с использованием этого ключа), свойство невосстанавливаемости ключа не подразумевает выполнения свойства стойкости к вычислению, поскольку не обязательно восстанавливать ключ, для того чтобы подделать MAC-код.

Задачу злоумышленника можно сформулировать следующим образом: без знания ключа k, вычислить новую пару "текст-MAC-код" (x', $h_k(x')$) для некоторого текста $x' \neq x_i$ по известным одной или более парам "текст-MAC" (x_i, $h_k(x_i)$).

Стойкость к вычислению должна обеспечиваться и в случае, когда тексты x_i, для которых доступны соотвествующие MAC-коды, заранее даны злоумышленнику, и в случае, когда он свободно выбирает эти тексты.

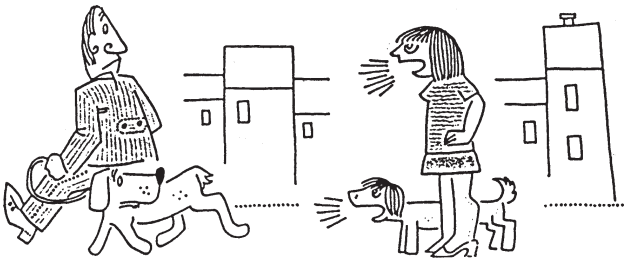
Можно выделить следующие атаки на MAC-коды:

1. Атака на основе открытого текста имеет место тогда, когда доступны одна или более пар "текст-MAC-код" (x_i, $h_k(x_i)$).

2. Атака на основе выбранного открытого текста имеет место тогда, когда доступны одна или более пар "текст-MAC-код" (x_i, $h_k(x_i)$) для выбранных злоумышленником текстов x_i.

3. Адаптивная атака на основе выбранного открытого текста имеет место тогда, когда открытый текст x_i выбирается злоумышленнием в условиях наличия у него некоторой дополнительной информации.

С точки зрения сертификации алгоритм формирования MAC-кода должен противостоять адаптивной атаке не зависимо от того, возможна или нет ее практичекая реализация, поскольку именно эта атака является наиболее опасной.



Атака на основе открытого текста...

Степень тяжести последствий, наступающих вследствие подделки MAC-кода, зависит от степени контроля злумышленником значений x, для которых возможна подделка MAC-кода. С этой точки зрения *возможны два типа подделки MAC-кода:*

- избирательная подделка, включающая множество атак, посредством которых злоумышленник способен породить новую пару "текст-MAC-код" для текста, который им же самим и выбран (или выбирались под его управлением). Заметим, что здесь выбранным значением является текст, для которого MAC-код подделан, тогда как в атаке на основе выбранного открытого текста текст из пары "текст-MAC-код" служит для анализа, с целью подделки MAC-кодов для других текстов;
- экзистенциальная подделка, включающая множество атак, посредством которых злоумышленник способен породить новую пару "текст-MAC-код", но без контроля над значением этого текста.

Наиболее опасной атакой, с точки зрения осуществления подделки MAC-кода, является восстановление ключа. При реализации всех типов атак, позволяющих подделать MAC-код, злоумышленник может выдавать поддельные сообщения за подлинные.

Бесключевые хеш-функции (034)

Выше уже отмечалось, что хеш-функции должны обладать как минимум двумя свойствами, а именно свойством сжатия и легкостью вычисления. Бесключевые хеш-функции, помимо этого, должны обладать дополнительными свойствами. Рассмотрим их.

Пусть h бесключевая хеш-функция, входными аргументами которой являются строки x и x', а выходными значениями строки y и y'. ***Бесключевая хеш-функция должна обладать следующими свойствами.***

1. Пусть дано некоторое y полученное по неизвестной входной строке x. Тогда вычислительно невозможно найти некоторое x' (x такое, что $h(x') = y$. По существу для всех заранее определенных выходных значений хеш-функции вычислительно невозможно найти какоелибо входное значение, которое отображается в заданное выходное значение. Такое свойство называется стойкостью к прообразу (preimage resistance) или односторонностью (one-way).

2. Стойкость ко второму прообразу (2nd-preimage resistance). Пусть известна некоторая строка x. Тогда вычислительно невозможно найти вторую строку (второй прообраз) x' (x такую, что $h(x) = h(x')$. Данное свойство носит название слабая стойкость к столкновению (weak collision resistance).

3. Стойкость к столкновению (collision resistance). Вычислительно невозможно найти любые две различные строки данных x и x' для которых $h(x) = h(x')$. В отличие от предыдущего свойства, стойкость к столкновению рассматривается в условиях, когда может осуществляться свободный выбор обеих входных строк. Данное свойство еще известно как сильная стойкость к столкновению (strong collision resistance).

В зависимости от свойств, которыми обладают хеш-функции, бесключевые хеш-функции можно разделить на два класса односторонние хеш-функции и свободные от столкновений хеш-функции.

Односторонняя хеш-функция *(ОСХФ)* или слабая односторонняя хеш-функция это хеш-функция, которая обладает свойствами сжатия, легкостью вычисления, стойкостью к прообразу и стойкостью ко второму прообразу.

Свободная от столкновения хеш-функция *(ССХФ)* или сильная односторонняя хеш-функции это хеш-функция, которая обладает свойствами сжатия, легкостью вычисления, стойкостью ко второму прообразу и стойкостью к столкновению.

Злоумышленник в общем случае при реализации атак на хеш-функции может решать следующие задачи:

- атака на ОСХФ — дан хеш-код y, найти сообщение x такое, что $y = h(x)$ или дана пара $(x, h(x))$, найти второе сообщение x' такое, что $h(x') = h(x)$.
- атака на ССХФ — найти любые два входных сообщения x и x' таких, что $h(x') = h(x)$.

Односторонние хеш-функции являются одним из фундаментальных криптографических примитивов и используются для обеспечения целостности данных в цифровых подписях, протоколах подтверждения знаний, схемах установления (выработки) общего ключа, а также в качестве элементов генераторов псевдослучайных чисел и во многих других приложениях.

В ходе использования хеш-функций были определены ***дополнительные практические свойства***, а именно:

- отсутствие корреляции между входными и выходными битами. Желательно, чтобы любой входной бит оказывал влияние на несколько выходных бит. Другими словами хеш-функция должна обладать достаточно хорошим лавинным эффектом;
- усложненная стойкость к столкновениям, которая заключается в том, что трудно найти два любых входных значения x и x' таких, что $h(x)$ и $h(x')$ отличались бы в небольшом количестве бит;
- частичная стойкость к прообразу и локальная односторонность, заключающиеся в том, что восстановление части сообщения является такой же сложной задачей, как и восстановление всего сообщения. Более того,

Хеш-фенкция имеет дополнительные практические свойства...

даже если известна часть входного сообщения, восстановление остатка также является трудоемкой задачей (например, для восстановления t неизвестных входных бит, необходимо выполнить в среднем 2^{t-1} операций хеширования.

Односторонние хеш-фунции определены в отдельном международном стандарте ISO/IEC 10118. Стандарт ISO/IEC 10118 состоит из четырех частей. Первые две части были опубликованы в 1994 году, третья часть — в 1998 году, а четвертая часть в настоящее время находится на стадии CD разработки.

Первая часть ISO/IEC 10118-1:1994 — общая часть, в которой водятся общие определения и понятия для других частей стандарта, а также модель итеративных хеш-функций.

ISO/IEC 10118-2:1994 определяет хеш-функции, использующие n-разрядный алгоритм блочного шифрования. В данной части описываются методы формирования хеш-кода на основе n-разрядного блочного шифра.

В третьей части ISO/IEC 10118-3:1998 определяются специализированные хеш-функции (dedicated hash function). Стандарт описывает три типа таких хеш-функций, а именно американский стандарт SHA-1, и европейский стандарт RIPEMD-128, RIPEMD-160.

Наконец в четвертой части стандарта ISO/IEC 10118-4 определены хеш-функции, использующие модулярную арифметику. Стандарт вводит две хеш-функции MASH-1 и MASH-2, которые используют модульное возведение в степень для построения хеш-кода.

Модель итеративных хеш-функций (034)

Все хеш-функции, определенные в ISO/IEC 10118, как и большинство бесключевых хеш-функций построены на основе итеративной модели. На рис. 9.4 представлена общая модель итеративной хеш-функции.

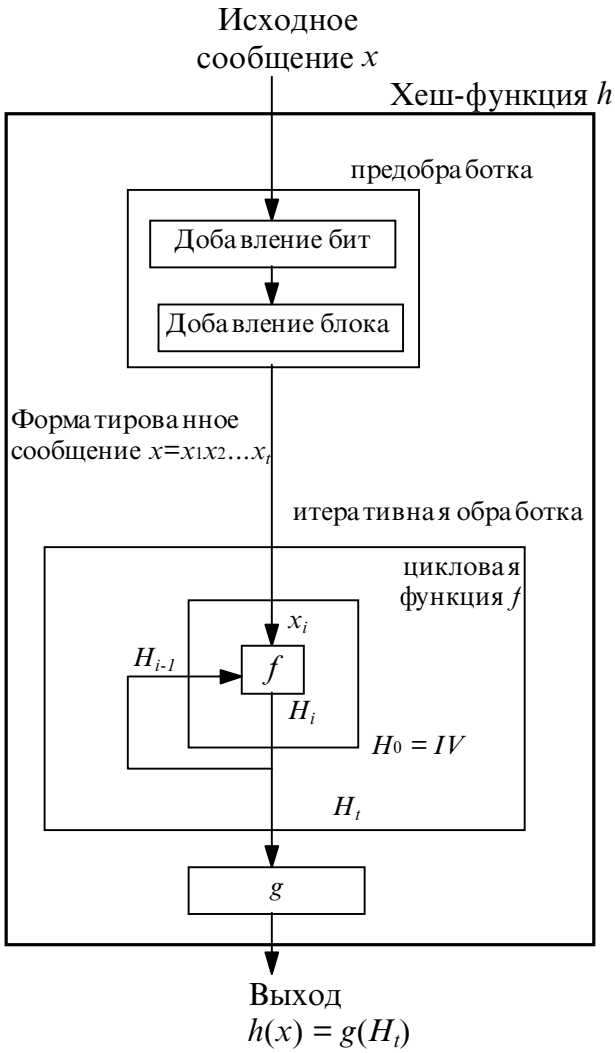


РИС. 9.4. *Итеративная модель хеш-функции*

Согласно данной модели формирование хеш-кода осуществляется следующим образом. На вход хеш-функции h поступает исходное сообщение x — строка данных произвольной длины. Данная строка представляется в виде последовательности r-разрядных блоков x_i. Перед обработкой, по необходимости, последний блок дополняется до r битов. Кроме того, из соображения повышения стойкости, исходная строка может быть дополнена новым r-разрядным блоком, который, например, может содержать двоичное представление числа, характеризующего длину исходного сообщения x. Стандарт ISO/IEC 10118 не определяет методы дополнения, но предполагается, что могут быть использованы методы, определенные в ISO/IEC 9797.

Основой хеш-функции является так называемая цикловая функция $f(x,y)$ или функция сжатия, которая, в общем случае, берет на вход две строки x и y, длиной соответственно r и s разрядов и формирует на выходе s-разрядную строку. Каждый блок x_i служит входным аргументом для цикловой функции. Другим аргументом является s разрядное промежуточное значение (переменная сцепления), полученное на предыдущем шаге хеширования. Пусть H_i обозначает частный результат хеширования на i-ом шаге (итерации), тогда общая модель итеративной хеш-функции со входом $x = x_1x_2 \ldots x_t$ может быть описана следующими соотношениями:

$$H_0 = IV;$$
$$H_i = f(x_i, H_{i-1}),\ 1 \le i \le t;$$
$$h(x) = g(H_t),$$

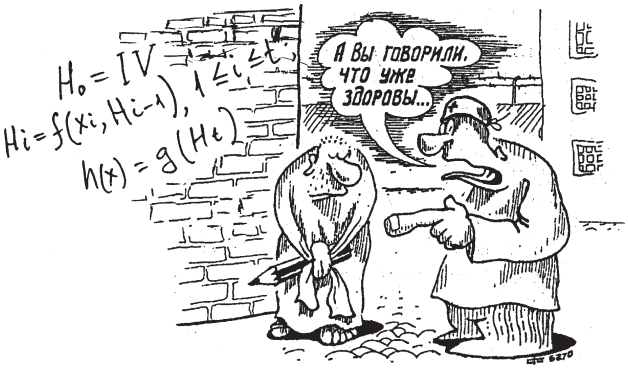
где H_{i-1} — s-разрядная переменная сцепления между i-1-ым и i-ым шагами обработки; H_0 — вектор инициализации.

На выходе цикловой функции f формируется s-разрядный хеш-код. При необходимости он может быть усечен до заданной длины посредством дополнительной обработки $g(H_t)$. Чаще всего $g(H_t) = H_t$. Таким образом, для формирования хеш-кода в соответствии с итеративной моделью хеш-функции по ISO/IEC 10118 необходимо выбрать следующие параметры:

- r — длина блока данных;
- s — длина выхода цикловой функции, так называемая переменная сцепления, которая определяет максимально возможную длину вырабатываемого хеш-кода;
- IV — s-разрядный вектор инициализации;
- $LH \le s$ — длина хеш-кода.

Специализированные хеш-функции (034)

Специализированные хеш-функции (Customized hash functions или dedicated hash functions) специально разработаны только для целей хеширования и оптимизированы для выполнения данной задачи. Третья часть стандарта ISO/IEC 10118-3 определяет три специализированные хеш-функции, а именно функции

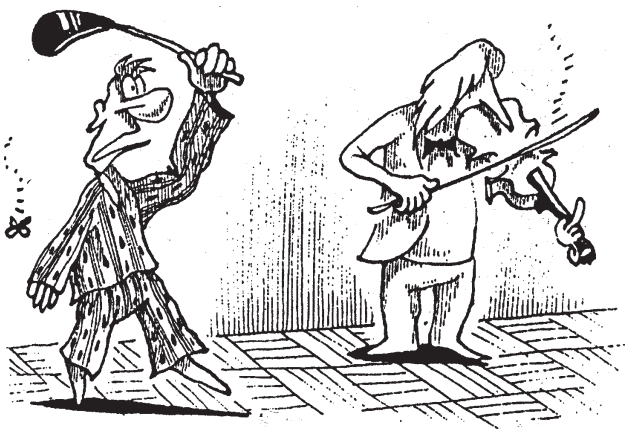


Основной хеш-функции является функция спасателя...

RIPEMD-128, RIPEMD-160 и SHA-1. Данные хеш-функции основаны на использовании принципов построения, заложенных в хеш-функции семейства MDx (MD2, MD4,MD5), которые специально разрабатывались для реализации на 32-разрядных ЭВМ. Алгоритм MD4 был предложен Р. Райвестом в 1990 году, а в 1991 тот же автор предложил модифицированную версию алгоритма MD5. В настоящее время хеш-функции MD4, MD5 являются наиболее распространенными в практических приложениях хеш-функциями. Однако некоторые их недостатки не позволили стандартизировать их.

Европейский консорциум RIPE, опираясь на свои исследования свойств этих алгоритмов предложил усиленную версию MD4, которая получила название RIPEMD. Хеш-функция RIPEMD по сути состоит из двух параллельно работающих и модифицированных функций MD4 (т.е. функция имеет две линии). Суть модификации заключается в изменении аргументов операторов циклических сдвигов и порядка следования на вход циклов хеш-функции слов хешируемого сообщения. Параллельные линии, кроме того, отличаются использованием различных констант. Версии RIPEMD-128 и RIPEMD-160 вырабатывают хеш-код длиной, соответственно, 128 и 160 бит.

Другой альтернативой алгоритмам MDx является алгоритм SHA-1, разработанный совместно Агентством национальной безопасности США и NIST и принятый в качестве американского национального стандарта (FIPS 180-1).



Циклический сдвиг аргументов...

Цифровые подписи (004)

Одним из механизмов, обеспечивающих реализацию функций аутентификации, целостности и причастности является механизм цифровой подписи.

Определение

Цифровая подпись (ЦП) представляет собой строку данных, которая зависит от некоторого секретного параметра (ключа), известного только подписывающему лицу, и от содержания подписываемого сообщения, представленного в цифровом виде. Таким образом, цифровая подпись связывает сообщение с некоторым порождающим или подписывающим его объектом.

Цифровым подписям посвящено несколько стандартов. Наиболее распространенными стандартами являются:

- международный стандарт ISO/IEC 9796 стандартизирует ЦП с восстановлением сообщения (digital signature with massage recovery);
- международный стандарт ISO/IEC 14888 стандартизирует ЦП с добавлением (digital signature with appendix);
- американский национальный стандарт цифровой подписи DSA (FIPS 186);
- российский национальный стандарт цифровой подписи ГОСТ Р 34.10;
- стандарт на ЦП PKCS #1;
- стандарт на цифровые подписи с добавлением и восстановлением сообщения IEEE 1363.

Большое количество нормативных документов еще раз указывает на то что цифровая подпись является одним из наиболее важных механизмов безопасности.

Общие определения и классификация схем цифровых подписей (034)

Для описания процессов обработки информации с использованием механизмов ЦП воспользуемся следующей терминологией.

1. ***Алгоритм генерации ЦП*** — это метод формирования ЦП.

2. ***Алгоритм проверки (верификации) ЦП*** — метод проверки того, что подпись является аутентичной, т.е. действительно создана конкретным объектом и не модифицирована при передаче.

3. ***Схема ЦП (или механизм ЦП)*** — совокупность взаимосвязанных алгоритмов генерации и верификации цифровой подписи.

4. ***Процесс (процедура) наложения ЦП*** — совокупность математического алгоритма генерации ЦП и методов представления (форматирования) подписываемых данных.

5. ***Процесс (процедура) снятия ЦП*** — совокупность алгоритма верификации ЦП и методов восстановления данных.

Для построения схемы ЦП необходимо определить два алгоритма: алгоритм генерации ЦП и алгоритм верификации ЦП. Алгоритм верификации доступен для всех потенциальных получателей подписанных сообщений, в то время как алгоритм генерации ЦП известен только подписывающему лицу, которое для некоторого сообщения $m \in \mathbf{M}$ определяет соответствующую подпись $s \in \mathbf{S}$. Верификатор, получив пару (m, s) и некоторую открытую информацию о подписывающем лице, применяет соответствующий алгоритм верификации ЦП. Данный алгоритм выдает двоичный результат: «да» если подпись верна (аутентична) и «нет» в противном случае.

Существующие на сегодняшний день схемы ЦП делятся на два класса (рис. 9.5):

- схемы ЦП с восстановлением сообщения;
- схемы ЦП с добавлением.

В ***схемах ЦП с восстановлением сообщения*** всё или часть подписанного сообщения может быть восстановлена непосредственно из цифровой подписи. Таким образом, на вход алгоритма верификации поступает лишь цифровая подпись s.

В ***схемах ЦП с добавлением*** цифровая подпись присоединяется к сообщению и в таком виде отправляется адресату. Для верификации такой ЦП необходимо иметь и подпись s и соответствующее сообщение m.

Каждая из этих схем может являться ***детерминированной*** или ***рандомизированной***. Применение детерминированных схем характеризуется тем, что цифровая подпись одной и той же входной строки данных приводит к формированию одинаковых цифровых подписей. В рандомизированной схеме при генерации подписи используется некоторый случайный параметр (число), что приводит формированию различных подписей для одинаковых входных строк (при использовании одних и тех же ключей). В рандомизированных схемах необходимо обеспечить непредсказуемость случайных чисел.

В свою очередь детерминированные схемы делятся на схемы ЦП одноразового применения (one-time) и схемы ЦП многократного применения (multiple-use).

Модель цифровой подписи с добавлением (034)

Схема ЦП с добавлением является наиболее распространенной схемой в практических приложениях. К этому классу ЦП относятся такие схемы подписи как DSA, ElGamal, Schnorr, ГОСТ 34.10. Рассмотрим модель схемы ЦП с добавлением.

Пусть необходимо подписать некоторое сообщение произвольной длины $m \in \mathbf{M}$, где $\mathbf{M}$ − пространство сообщений. Предварительно сообщение m хешируется с использованием односторонней и свободной от коллизий хеш-функции

$$\tilde{m} = h(m),\ \tilde{m} \in \mathbf{M}_h,$$

где $\mathbf{M}_h$ − пространство хеш-кодов.

Напомним, что свойство односторонности означает, что по данной произвольной строке y (хеш-коду) вычислительно невозможно найти двоичную строку x такую, что $h(x) = y$ несмотря на то, что такая строка существует и, в общем случае не одна. Свойство свободности от коллизий означает, что вычислительно невозможно най-

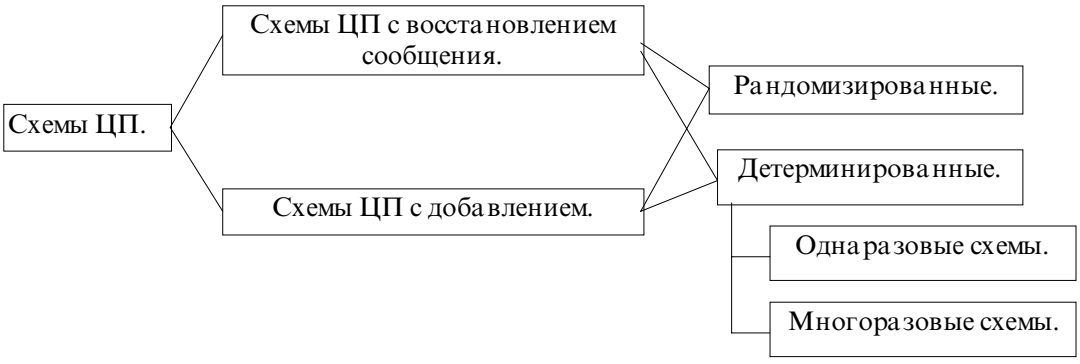


РИС. 9.5. *Классификация схем ЦП.*

ти две строки $x \neq x'$ таких, что $h(x) = h(x')$, несмотря на то, что такие пары строк существуют.

Используя личный ключ $k \in \mathbf{K}$, где $\mathbf{K}$ − пространство ключей , выбранный алгоритм генерации подписи SIG (•) и хеш-код сообщения $\tilde{m}$ отправитель генерирует подпись $s \in \mathbf{S}$, где $\mathbf{S}$ − пространство подписей

$$s = SIG_k(\tilde{m}).$$

Сообщение m и добавляемая к нему подпись s отправляется получателю.

Получатель верифицирует подпись следующим образом. он получает в своё распоряжение аутентичную копию ключа верификации $k_v \in \mathrm{K}'$. Затем по полученному сообщению m вычисляет хеш-код $\tilde{m} = h(m)$ и, используя алгоритм верификации VER(•), принимает решение относительности истинности или ложности подписи:

$$VER_{k_v}(\tilde{m}, s) \rightarrow \{\text{истина, ложь}\}.$$

На рисунке 9.6 приведена схема модели ЦП с добавлением.

К ЦП с добавлением предъявляются следующие общие требования:

1) для любого ключа $k \in \mathbf{K}$ алгоритм генерации SIG_k (•) должен эффективно вычисляться;

2) ключ верификации подписи $k_v \in \mathbf{K}'$ должен быть защищен от подделки и для любого k_v алгоритм верификации подписи также должен эффективно вычисляться;

3) любому субъекту, за исключением отправителя, вычислительно невозможно найти $m' \neq m$, $m' \in \mathbf{M}$ и $s' \neq s$, $s' \in \mathbf{S}$ такие, что применение алгоритма верификации $VER_{k_v}(\tilde{m}, s')$, где $= h(m')$ даст значение «истина».

Модель цифровой подписи с восстановлением сообщения (004)

В схемах ЦП с восстановлением сообщения подписываемое сообщение может быть восстановлено непосредственно из подписи. На практике ЦП с восстановлением используются для работы с короткими

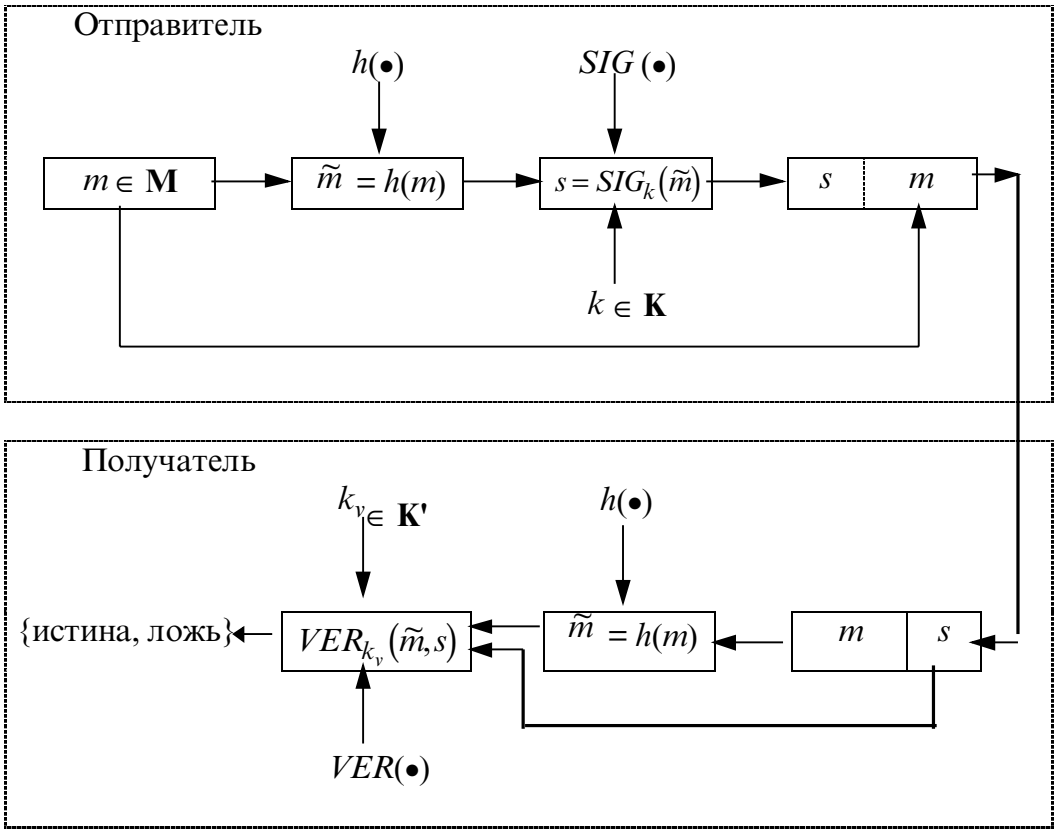
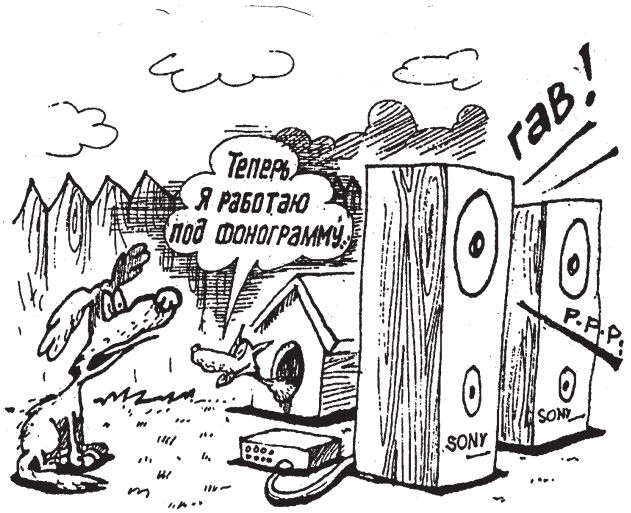


РИС. 9.6. *Цифровая подпись с добавлением.*

сообщениями. Примерами таких схем являются схемы подписи RSA, Rabin, Nuberg-Rueppel (NR-схема).

Пусть отправитель формирует цифровую подпись $s \in \mathbf{S}$ для некоторого сообщения $m \in \mathbf{M}$. При этом в дальнейшем получатель может восстановить сообщение m из s. Сначала в сообщение m вносится избыточность, путем применения функции избыточности R

$$\tilde{m} = R(m), \tilde{m} \in \mathbf{M}_R,$$



Модель с восстановлением сообщения...

где $\mathbf{M}_R$ – пространство избыточных сообщений или пространство избыточности.

Затем формируется цифровая подпись согласно выражению

$$s = SIG_k(\tilde{m}).$$

Цифровая подпись s отправляется получателю.

Для верификации получатель должен получить аутентичный ключ верификации $k_v \in \mathbf{K}'$ отправителя и вычисляет избыточное сообщение из подписи согласно выражению

$$\tilde{m} = VER_{k_v}(s).$$

В случае если $\tilde{m} \notin \mathbf{M}_R$, то подпись отвергается как недействительная. после этой проверки восстанавливается оригинальное сообщение m из $\tilde{m}$ путем вычисления $R^{-1}(m)$.

На рисунке 9.7 представлены процедуры наложения и верификации ЦП с восстановлением сообщения.

К ЦП с восстановлением предъявляются следующие требования:

1) для любого ключа $k \in \mathbf{K}$ алгоритм генерации $SIG_k(\bullet)$ должен эффективно вычисляться;

2) ключ верификации подписи $k_v \in \mathbf{K}'$ должен быть защищен от подделки и для любого k_v алгоритм верификации подписи $VER_{k_v}(\bullet)$ также должен эффективно вычисляться;

3) любому субъекту, за исключением отправителя, вычислительно невозможно найти $s' \neq s$, $s' \in \mathbf{S}$ такое, что $VER_{k_v}(s) \in \mathbf{M}_R$.

В цифровой подписи с восстановлением функция избыточности R и её инверсия R^{-1} являются открытыми. Выбор соответствующей функции R является критичной задачей с точки зрения обеспечения необходимой стойкости схемы ЦП. В ЦП с восстановлением сообщения мощность пространства сообщений $|\mathbf{M}|$ меньше мощности пространства подписей $|\mathbf{S}|$. Если бы $|\mathbf{M}| = |\mathbf{S}|$, то поиск подписи s по соответствующему m был бы тривиален. Функция избыточности осуществляет обратимое отображение сообщения m в избыточное сообщение $\tilde{m} \in \mathbf{M}_R$. В свою очередь алгоритм генерации подписи $SIG_k(\bullet)$ применяется к пространству подписываемых сообщений M_S, причем $|\mathbf{M}_S| > |\mathbf{M}_R|$, т.е. $\mathbf{M}_R \subseteq \mathbf{M}_S$. Считается, что хорошая функция избыточности должна обеспечивать соотношение мощностей такое, что $|\mathbf{M}_R| / |\mathbf{M}_S| = \left(\frac{1}{2}\right)^n$, где n – длина подписываемого сообщения. В схемах ЦП с восстановлением сообщения функция избыточности является средством обнаружения модификации цифровой подписи, т.е. является аналогом хеш-функции, только с обратным знаком. Функция избыточности для любого сообщения m порождает и вводит уникальную избыточность, которая зависит от содержания информации в сообщении. Функция R должна быть легко вычислима и чаще всего зависит от типа алгоритма генерации подписи. Например, функция введения избыточности предложенная в международном стандарте ISO/IEC 9796 применима только с преобразованиями RSA и Rabin.

Любая схема с восстановлением сообщения может быть преобразована в схему ЦП с добавлением путем хеширования сообщения, а затем применения схемы ЦП с восстановлением сообщения к полученному хеш-коду. В этом случае алгоритм верификации требует наличия и самого сообщения. Процедура наложения подписи для такого варианта представлена на рис. 4.

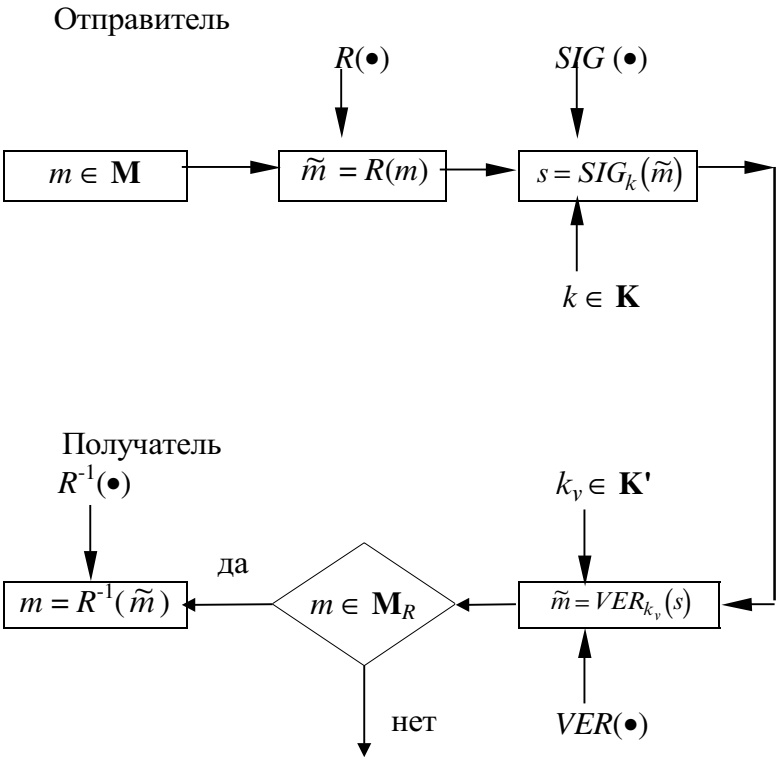


РИС. 9.5. *Цифровая подпись с восстановлением сообщения.*

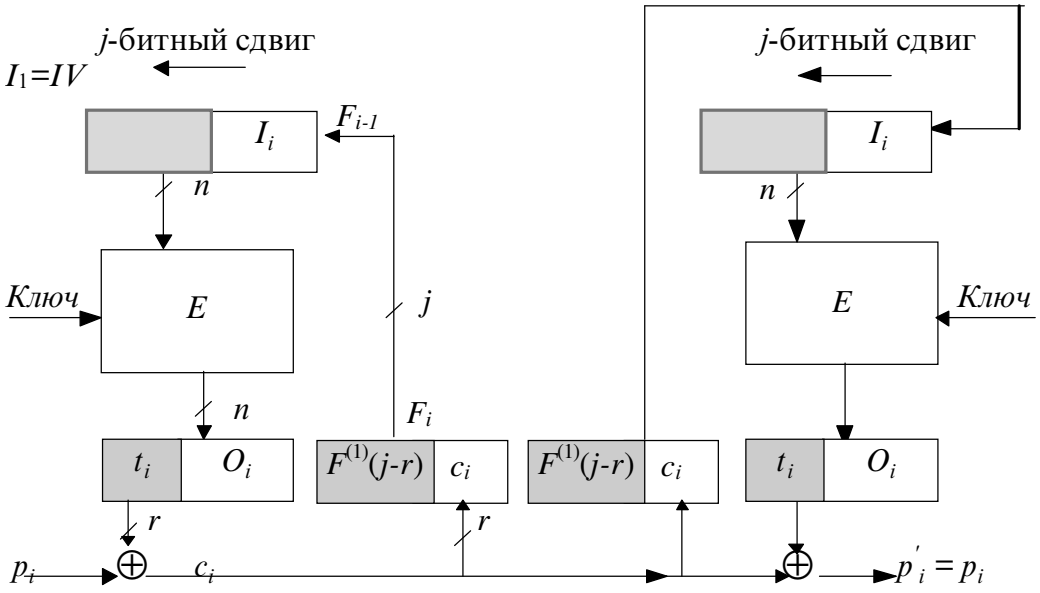


РИС. 9.6. *Комбинированная цифровая подпись.*

В данном случае к функции избыточности R предъявляются не такие жесткие требования и она может быть просто взаимооднозначной функцией отображения из множества $\mathbf{M}_h$ во множество $\mathbf{M}_S$.

Механизмы неотказуемости (причастности) (004)

Под причастностью (неотказуемостью) понимают предотвращение возможности отказа одним из реальных участников обмена сообщениями от факта его полного или частичного участия в передаче данных.

Общая модель обеспечения причастности включат в себя следующие основные стороны:

- источник сообщения;
- получатель сообщения;
- доверительная третья сторона, которая может выступать в роли полномочного органа доставки или нотариуса.

Реализация механизмов причастности стандартизирована в международном стандарте ISO/IEC 13888.

Модель взаимоотношений, в условиях которой взаимодействуют эти три стороны, можно назвать моделью взаимного недоверия, т. е. все стороны не имееют оснований доверять друг другу и считают, что каждая из них может совершить попыку обмана относительно другой.

В условиях такой модели можно выделить следующие типы обманов.

1. Обман состороны получателя сообщений. Получатель, в случае отсуствия сообщения со стороны источника, может попытаться совершить обман, путём формирования некотрого ложного сообщения и регистрации его как сообщения, переданного ему источником. Успех обмана будет заключаться в том, что если источник попытается отказаться от этого ложного сообщения, то нотариус вынесит решение против него. Используя такую стратегию получатель может приписать ложное сообщение источнику.

При наличии сообщения, получатель может модифицировать полученное сообщение. Наконец получатель может просто отказаться от факта получения сообщения, хотя на самом деле его получил.

2. Обман со стороны источника сообщений. Источник может попытаться обмануть получателя, послав ему некторое собщение, которое естественно получатель сообщения принимает правильно, а затем источник отрицает факт передачи сообщения.

3. Наконец, ***доверительная третья сторона*** также может совершать обманные действия. ДТС, используя свою привилегированную информацию об алгоритмах, которые используются в механизмах причастности, может сформировать и послать ложное сообщение получателю от имени источника, т.е. совершить имитацию сообщения. С другой стороны ДТС может перехватить истинное сообщение, переданное санкционированным источником и, используя дополнительную информацию, подменить его ложным сообщением.

Исходя из возможных типов обманов, стандарт определяет восемь услуг неотказуемости. ***Наиболее важными среди них являются следующие четыре услуги:***

- ***неотказуемость происхождения*** (Non-repudation of origin) — защита от обмана со стороны источника сообщения, который ложно отрицает факт посылки сообщения;
- ***неотказуемость приёма на доставку*** (Non-repudation of submission) — защита от обмана со стороны полномочного органа доставки сообщения, который ложно отрицает факт приёма сообщения от источника;
- ***неотказуемость передачи*** (Non-repudation of transport) — защита от обмана со стороны полномочного органа доставки, который ложно отрицает факт доставки сообщения полученного от источника;
- ***неотказуемость получения*** (Non-repudation of delivery) — защита от обмана со стороны получателя сообщения, отрицающего факт получения сообщения.

Каждая из этих услуг обеспечивается путём предоставления доказательств причастности соответствующим сторонам. Услуга неотказуемости происхождения требует предоставления доказательства причастности получателю сообщений. Остальные услуги требуют предоставления доказательств причастности источнику сообщения.

Во второй части стандарта ISO/IEC 13888-2 определяются механизмы неотказуемости на основе использования методов симметричного шифрования и методов формирования MAC-кодов с обязательным привлечением

Обман со стороны источника...

доверительной третьей стороны. При применении данных механизмов общая модель причастности несколько модифицируется. Источник и приемник сообщений не доверяют друг другу, но оба доверяют доверительной третьей стороне. Таким образом, в такой модели можно реализовать услуги неотказуемость происхождения и неотказуемость получения сообщения.

Рассмотрим один из механизмов неотказуемости, основанный на вовлечении ДТС, который обеспечивает неотказуемость происхождения сообщения.

Пусть источник сообщения A посылает сообщение m приемнику B, и пусть A и B доверяют некоторой третьей стороне (ДТС). Механизм доказательства неотказуемости происхождения сообщения выглядит следующим образом.

Механизм доказательства неотказуемости происхождения собщения (034)

Резюме. Источник собщения *A* доказывает свою причастность к происхождению собщения *m* получателю *B* через использование пяти-проходного протокола с привлечением ДТС.

1. ***Установка системных параматеров.*** Источник сообщения *A*, получатель сообщения *B* и ДТС обладают уникальными идентификаторами ID_A, ID_B, $ID_{\text{ДТС}}$, соответственно.

Источник сообщения *A*, получатель сообщения *B* и ДТС обладают общим алгоритмом формирования кода подлинности сообщения *d* с использованием ключа *k* – $\text{MAC}_k(d)$ и общей бесключевой хеш-функцией $h(m)$. (Выбор MAC-кода и $h(m)$ может осуществляться на основе ISO/IEC 9797, ISO/IEC 10118).

Источник сообщения *A* и ДТС обладают общим секретным ключом *a*, получатель сообщения *B* и ДТС обладают общим секретным ключом *b*, ДТС обладает секретным ключом *k*.

2. ***Предварительная обработка сообщения.*** Источник сообщения *A* путем конкатенции формирует строку данных *z*:

$$z = ID_A \parallel ID_B \parallel ID_{\text{ДТС}} \parallel T \parallel h(m),$$

где *T* – метка времени.

3. ***Протокол обмена сообщениями.*** При каждой посылке *m* осуществляется обмен следующими сообщениями.

(1) $A \rightarrow$ ДТС: $z \parallel \text{MAC}_a(z)$.
(2) ДТС $\rightarrow A$: $z \parallel \text{MAC}_k(z) \parallel \text{MAC}_a(z \parallel \text{MAC}_a(z))$.
(3) $A \rightarrow B$: $m \parallel z \parallel \text{MAC}_k(z)$.
(4) $B \rightarrow$ ДТС: $z \parallel \text{MAC}_k(z) \parallel \text{MAC}_b(z \parallel \text{MAC}_k(z))$.
(5) ДТС $\rightarrow B$: $\text{PON} \parallel z \parallel \text{MAC}_k(z) \parallel \text{MAC}_b(\text{PON} \parallel z \parallel \text{MAC}_k(z))$.

Здесь PON – флаговый бит (положительный или отрицательный), указывающий является ли действительной информация неотказуемости (признак неотказуемости).

4. ***Действия сторон по протоколу.***

(1). Источник сообщения *A* в составе строки *z* отправляет образ собщения и метку времени ДТС, а также код подлинности строки *z*, вычисленный на общем с ДТС ключе *a*.

(2). После получения от ДТС сообщения (2) источник сообщения *A* убеждается в том, что это сообщения получено именно от ДТС и что $\text{MAC}_k(z)$ вычислен именно по *z* путем вычисления значения $\text{MAC}_a(z \parallel \text{MAC}_a(z))$ и сравнения его с полученным от ДТС.

(3). Код подлинности $\text{MAC}_k(z)$ используется источником собщения *A* как доказательство его причастности к происхождению сообщения *m* предоставляемое получателю сообщения *B*.

(4). После получения от *B* собщения (4) ДТС убеждается в том, что сообщения получено от *B*, путем вычисления $\text{MAC}_b(z \| \text{MAC}_k(z))$ и сравнения его с полученным от *B*. Затем ДТС проверяет, что присланный $\text{MAC}_k(z)$ действительно вычислен по *z* и формирует признак неотказуемости PON.

(5). После получения последнего сообщения *B* должен сохранить $z \| \text{MAC}_k(z)$ как доказательство того, что *A* действительно посылал сообщение *m* стороне *B*.

Третья часть стандарта ISO/IEC 13888-3 описывает механизмы построения и использования цифровых подписей для обеспечения различных услуг неотказуемости.

Например, признак неотказуемости получения сообщения определяется как подпись получателя под строкой данных, которая содержит следующую информацию

$$z = \{ID_{\text{источника}}, ID_{\text{получателя}}, \text{метка времени}, h(m)\}.$$

Для того, чтобы обеспечить услугу неотказуемости получения, получатель сообщения, по запросу источника сообщения, в обязательном порядке должен будет предоставить признак неотказуемости получения. Заметим, что при использовании механизмов на основе цифровых подписей из модели причастности можно исключить ДТС, либо ДТС будет выполнять роль арбитра, причем арбитру также могут не доверять.

В информативном приложении ISO/IEC 13888-3 также описывается использование ДТС в обеспечении услуги установки метки времени. Суть услуги заключается в том, что в модель причастности вводится ДТС, которая добавляет метку времени и свою подпись к данным, предоставляемым запрашивающей стороне. Такими данными, например, может быть предварительно подписанный признак неотказуемости. Использование услуги отметки времени особенно актуально тогда, когда цифровые подписи, а следовательно и

признаки неотказуемости, имеют большой срок легитимности (например при использовании услуг архивирования данных). Добавление метки времени третьей доверительной стороной позволяет обеспечить защиту от последующей отмены по истечению срока общего (или секретного) ключа, используемого для подписи признака неотказуемости.

Режимы работы блочных алгоритмов шифрования (034)

Режимы работы для алгоритма DES стандартизованы в США в 1980 году (FIPS 81) и в 1983 году (ANSI X3.106-1983). Эти режимы работы являются рекомендуемыми способами использования DES для шифрования данных. Сначала в ISO также начали проводить работы по разработке соответствующих международных стандартов для режимов работы DES. Когда же в ISO работы по DES были прекращены, работы по стандартизации режимов работы продолжались, но теперь по отношению к любому алгоритму блочного шифрования. В результате появилось два стандарта: ISO 8372: 1987 – режимы работы для 64-разрядных алгоритмов блочного шифрования и ISO/IEC 10116:1997 – рабочие режимы *n*-разрядных алгоритмов блочного шифрования (для любого *n*). Кроме того в 1991 году был представлен стандарт ISO/IEC 10118, на основе которого и разрабатывался ISO/IEC 10116, который является второй редакцией, содержащей расширенную версию *CFB* режима.

Все упомянутые выше стандарты описывают четыре рабочих режима:

- режим электронной кодовой книги *(Electronic Code Book Mode, ECB)*;
- режим сцепления блоков шифр текста *(Cipher Block Chaining Mode, CBC)*;
- режим обратной связи по шифртексту *(Ciphertext Feed Back Mode CFB)*;
- режим обратной связи по выходу *(Output Feed Back Mode, OFB)*.

Рассмотрим эти режимы более подробно. Все дальнейшие рассуждения в основном опираются на стандарт ***ISO/IEC 10116:1997 – Информационные технологии – Методы обеспечения безопасности – Режимы работы n-битного блочного шифра (ISO 8372 является частным случаем ISO/IEC 10116).***

Далее мы будем использовать такие обозначения. Символ *E* будем использовать для обозначения операции шифрования *n*-битного блочного шифра, где *n* – количество бит в блоках открытого и закрытого текстов, а символ *D* будем обозначать операцию дешифрования для того же шифра. Преобразование открытого текста *P* в закрытый текст *C* осуществляется по формуле:

$$C = E_k(P),$$

где *C* — *n*-битный блок шифртекста; *K* — ключ; *P* — *n*-битный блок открытого текста.

Аналогичным образом определим обратное преобразование:

$$P = D_k(C),$$

Также справедливо соотношение вида:

$$P = D_k(E_k(P)).$$

Режим электронной кодовой книги

Схема режима представлена на рис. 9.9.

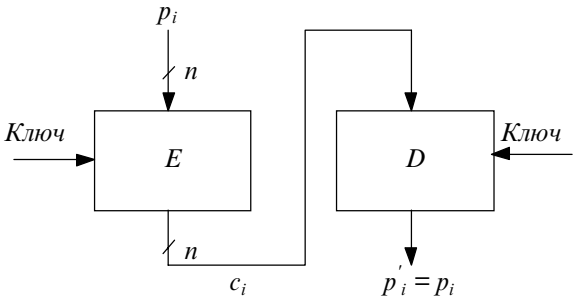


РИС. 9.9. *Режим электронной кодовой книги*

На вход алгоритма поступает *k*-битный ключ *K* и *n*-битные блоки открытого текста $P = p_1, p_2, ..., p_t$.

На выходе формируются *n*-битные блоки шифртекста $C = c_1, c_2, ..., c_t$.

Шифрование: для $1 \le i \le t$ выполнить операцию шифрования

$$c_i = E_k(p_i).$$

Дешифрование: для $1 \le i \le t$ выполнить операцию дешифрования

$$p_i = D_k(c_i).$$

Отметим следующие ***свойства ECB режима.***

1. ***Шифрование*** идентичных открытых текстов. Блочные шифры в таком режиме не обеспечивают скрытия шаблонов данных – идентичные блоки открытого текста (при одном и том же ключе) преобразуются в идентичный шифртекст.

2. ***Эффект сцепления блоков.*** Блоки шифруются независимо друг от друга. Переупорядочивание блоков шифртекста приведет к переупорядочиванию блоков открытого текста. Поскольку блоки шифртекста независимы, умышленная подстановка блоков в режиме *ECB* (например, вставка часто встречающегося блока) не оказывает влияния на дешифрование смежных блоков.

3. ***Размножение ошибок при дешифровании***. Один и более ошибочных бит в одном блоке шифртекста оказывает влияние на дешифрование только этого блока. Результат дешифрования блока с ошибками является случайной величиной, с около 50-ти процентной восстанавливаемостью битов открытого текста.

Из перечисленных свойств вытекает и использование *ECB* режима. *ECB* режим не рекомендуется использовать для шифрования сообщений больших чем один блок, или, если повторно используются ключи, для шифрования более чем одного одноблокового сообщения. Отчасти секретность ключа может быть обеспечена путем включения случайных дополнительных бит в каждый блок.

Режим сцепления блоков шифра (034)

Режим сцепления блоков шифра включает использование *n*-разрядного вектора инициализации, обозначаемого *IV*. В схему добавляется регистр или буфер обратной связи, в который сначала записывается значение *IV*, а затем последующие значения блоков шифртекста. На рис. 2 показана схема *CBC*.

На вход алгоритма поступает *k*-битный ключ *K*, *n*-битный *IV*, последовательность *n*-битных блоков открытого текста $P = p_1, p_2, ..., p_i, ..., p_t$.

Последовательность блоков шифртекста $C = c_1, c_2, ..., c_t$ вычисляется по правилу:

$$c_i = E_k(p_i \oplus c_{i-1}); \quad i = 1,...,t; \quad c_0 = IV$$

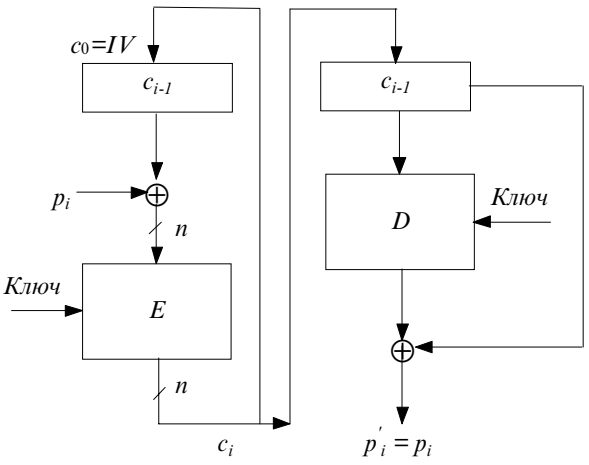
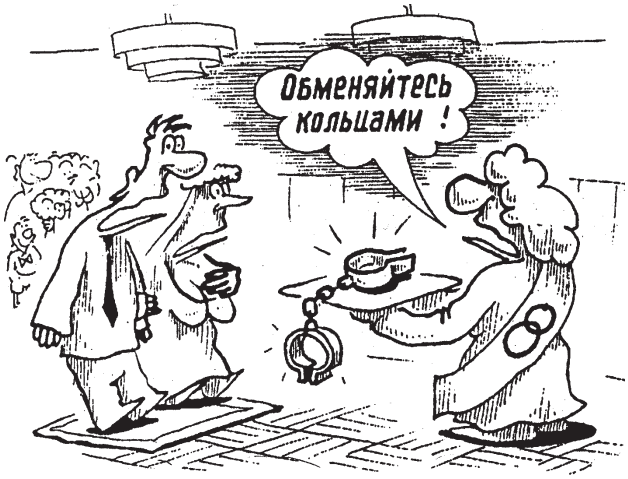


РИС. 2. *Режим сцепления блоков шифра*

Дешифрование осуществляется как:

$$p_i = c_{i-1} \oplus D_k(c_i); \quad i = 1,...,t; \quad c_0 = IV$$



Режим сцепления блоков...

Свойства режима

1. ***Шифрование идентичных открытых текстов.*** Идентичные блоки шифртекста формируются тогда, когда шифруется один и тот же открытый текст на одних и тех же ключе и векторе инициализации. Изменение вектора инициализации, ключа или первого блока открытого текста (например, используя счетчик или случайную величину) приводит к различным шифртекстам.

2. ***Эффект сцепления блоков.*** Механизм сцепления приводит к тому, что блок шифртекста c_i зависит от блока открытого p_i и всех предшествующих блоков открытого текста. В этом случае входная зависимость между собо всех предшествующих блоков открытого текста содержится в значении предыдущего блока шифртекста. Следовательно, переупорядочивание порядка следования блоков шифртекста оказывает влияние на дешифрование. Для того, чтобы правильно дешифровать блок шифртекста необходимо иметь правильный предыдущий блок шифртекста.

3. ***Размножение ошибок пи дешифровании.*** Единичная битовая ошибка в блоке шифртекста c_i оказывает влияние на дешифрование как минимум двух блоков c_i и c_{i+1} (поскольку p_i зависит от c_i и c_{i+1}). Блок p'_i восстановленный из c_i обычно является полностью случайным (50% битов ошибочны), в то время как восстановленный открытый текст p'_{i+1} будет иметь неправильных бит ровно столько, сколько их имеет блок c_i. Таким образом, противник может вызвать предсказуемые изменения бит в p'_{i+1}, изменяя соответствующие биты блока c_i.

4. ***Восстановление ошибок.*** *CBC* режим является реализацией самосинхронизирующего шифра или шифрования с автоключом по шифртексту в том смысле, что если имеет место ошибка в блоке c_i (включая потерю одного

или более входных блоков), но нет ошибок в c_{i+1}, c_{i+2}, то возможно корректное дешифрование p_{i+2}. Однакко, несмотря на то, что самосинхронизация возможна при наличии битовых ошибок в пределе блока (модификация битов), самосинхронизация при «потерянных» битах, являющихся причиной ошибки определения границ блока (ошибка целостности кадра или нарушение синхронизации), невозможна ни в *CBC* режиме ни в других режимах.

5. ***Размножение ошибок при шифровании.*** Хотя дешифрование в *CBC* режиме позволяет восстановить открытый текст при возникновении ошибок в блоках шифртекста, модификация блока открытого текста p_i ai a?aiy шифрования изменяет все последующие блоки шифртекста. Это усложняет (если не исключает) применимость режима сцепления для приложений, требующих случайного доступа (чтения/записи) к шифруемым данным. Альтернативой является применения режима *ECB*.

При использовании режима *CBC* режима,несмотря на то, что вектор инициализации *IV* е не является секретным, должна быть обеспечена его целостность, поскольку предумышленная модификация *IV* позволяет противнику выполнять предсказуемые изменения бит в восстанавливаемом первом блоке открытого текста. Использование секретных *IV* является одним из методов предотвращения таких модификаций. Однако, если требуется обеспечить целостность сообщения, должны использоваться другие механизмы, поскольку механизмы шифрования обычно гарантируют только конфиденциальность.

Режим обратной связи по шифртексту (034)

В *CBC* режиме обработка открытого текста осуществляется поблочно. Некоторые приложения требуют, чтобы *r*-битные части открытого текста были зашифрованы и переданы без задержки, для некоторого $r < n$ (часто $r = 1$ или $r = 8$). В этом случае может быть использован режим обратной связи по шифртексту.

В *CFB* режиме на вход подается *k*-битный ключ *K*, *n*-битный *IV*, последовательность *r*-битных блоков открытого текста $P = p_1, p_2, ..., p_u$, где $1 \le r \le n$. На выходе формируются *r*-битные блоки шифртекста $C = c_1, c_2, ..., c_u$.

Шифрование осуществляется следующим образом.

Вначале осуществляется запись в буфер обратной связи значения *IV*

$$I_1=IV.$$

Затем для всех $1 \le i \le u$ выполняются следующие операции:

1. Вычислить функцию шифрования

$$O_i = E_k(I_i).$$

2. Выбрать *r* крайних слева бит реультата шифрования O_i

$$t_i = O_i \sim r,$$

где $O_i \sim r$ означает операцию выборки.

3. Сформировать и передать *r*-битный блока шифртекста c_i

$$c_i=p_i \text{ E } t_i.$$

4. Сформировать переменнуь обратной связи из блока шифртекста, согласно выражению

$$I_{i+1}=2^r I_i \oplus c_i \bmod 2^n.$$

Эта операция эквивалентна сдвигу значения буферного регистра I_i на *r* бит вправо и записи в конец регистра *r*-битного блока c_i.

Дешифрование выполняется в соответствии с выражением:

$$P_i=c_i \oplus t_i; \quad i=1,..., u,$$

где t_i, O_i и I_i вычисляются согласно выражениям приведенным выше и $I_1= IV$.

На рисунке 3 представлена схема *CFB* режима.

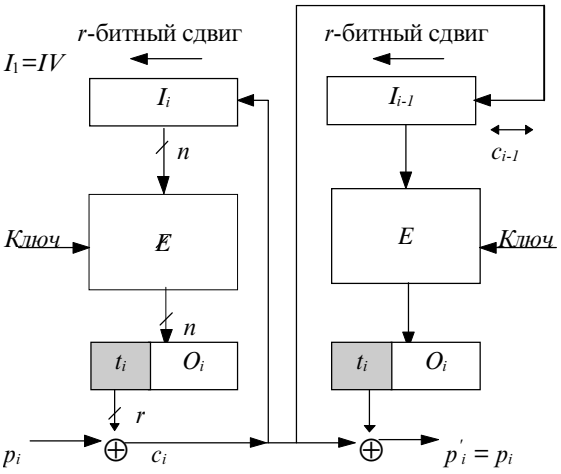


РИС. 3. *Режим обратной связи по шифртексту.*

Отметим, что фактически необходимо иметь еще и буфер обратной связи, для хранения c_{i-1}, который обычно выбирается равным *n*.

CFB режим обладает следующими свойствами

1. ***Шифрование идентичных открытых текстов.*** Как и в CBC режиме шифрования, изменение *IV* приводит к различным результатам шифрования одного и того же

открытого текста. Значение IV не обязательно должно быть секретным, хотя непредсказуемость IV является желательной в некоторых приложениях.

2. ***Эффект сцепления блоков.*** Аналогично с *CBC* режимом шифрования, механизм сцепления приводит к тому, что блок шифртекста c_i зависит от двух блоков открытого текста p_i и p_{i-1}. Следовательно, переупорядочивание блоков шифртекста оказывает влияние на дешифрование. Корректно дешифрование неискаженных блоков шифртекста требует получения $\lceil n/r \rceil$ неискаженных предыдущих блоков шифртекста, для того чтобы регистр обратной связи содержал верное значение.

3. ***Размножение ошибок при дешифровании.*** Одна или более ошибок в любом *r*-битном блоке шифртекста c_i, влияет на дешифрование следующих $\lceil n/r \rceil$ блоков шифртекста, то есть пока не будет закончена обработка *n* бит шифртекста, после которых ошибочный блок c_i будет выдвинут из регистра обратной связи. Восстановленный открытый текст p_i' будет отличаться от p_i точно в тех же позициях, в которых произошли ошибки в c_i. Другие некорректно восстановленные блоки открытого текста будут являться случайными векторами, то есть иметь 50% ошибочных бит. Таким образом, злоумышленник может осуществить предсказуемое изменения бит в p_i путем изменения соответствующих бит в c_i.

4. ***Восстановление ошибок***. *CFB* режим, как и *CBC* режим, самосинхронизирующийся, но требует получения $\lceil n/r \rceil$ блоков шифртекста для восстановления синхронизации.

5. ***Производительность***. Для $r < n$ производительность уменьшается в n/r раз (по сравнению с *CBC)*, так как каждое шифрование *E* обеспечивает формирование только *r* бит шифртекста на выходе.

Рассмотренный режим является режимом работы с *r*-битными входными символами и *r*-битной обратной связью. Этот режим схож с режимом *CFB,* стандартизированным для DES (FIPS 81 и ANSI X3.106) и обозначается как *CFB r*-битный символ /*r*-битная обратная связь. Стандарт ISO/IEC 10118:1991 определяет более общий *CFB* режим. При использовании этого режима может осуществляться обработка *r*-битных блоков открытого текста при *j*-битной обратной связи, причем *r* J *j*. Таким образом, для реализации режима необходимо выбрать величину обратной связи *j* ($1 \le j \le n$) и величину блока открытого текста *r* ($1 \le r \le j$).

На вход поступает *k*-битный ключ *K*, *n*-битный *IV*, последовательность *r*-битных блоков открытого текста $P = p_1, p_2, ..., p_u$.

На выходе алгоритма формируются *r*-разрядные блоки шифртекста $C = c_1, c_2, ..., c_u$.

Шифрование осуществляется в соответствии со следующими операциями

Перед шифрованием осуществляется запись в регистр обратной связи значения вектора инициализации $I_1 = IV$.

Для всех $1 \le i \le u$ выполнить следующие операции

1. Вычислить функцию шифрования:

$$O_i = E_k(I_i);$$

2. Выбрать *r* левых бит величины O_i:

$$t_i = O_i \sim r;$$

3. Сформировать блок шифртекста:

$$c_i = p_i + t_i;$$

4. Сформировать переменную обратной связи из блока шифртекста c_i путем его дополнения слева *j*-*r* единицами:

$$F_i = F^{(1)}(j\text{-}r) \,||\, c_i,$$

где $F^{(1)}(j\text{-}r)$ – блок из *j*-*r* единиц; || – операция конкатенции.

5. Выполнить сдвиг содержимого регистра I_i на *j*-бит влево и записать F_i в регистр:

$$I_{i+1} = 2^j I_i + F_i \bmod 2^n.$$

Отметим, что для $i = u$ операции 4 и 5 не выполняются.

Дешифрование выполняется в аналогичном порядке, за исключением того, что:

$$P_i = c_i \oplus t_i.$$

На рис. 9.11 представлено схематическое изображение функционирования алгоритма шифрования в *CFB* режиме по ISO/IEC 10118: 1991.

И, наконец, во второй редакции стандарта ISO/IEC 10116:1997 была предложена обобщенная версия CFB режима. Стандартизированный режим обеспечивает "конвеерную обработку данных".

В ранее рассмотренных версиях CFB режима результат шифрования одного блока открытого текста является входом для шифрования следующего блока. Это означает, что невозможны "конвеерные" вычисления, то есть невозможно приступить к шифрованию одного блока до окончания обработки предыдущего блока. Чтобы исключить этот недостаток, в новую версию CFB режима введен уже m-разрядный буфер обратной связи, где $2n \le m \le n$. Также для работы необходим m-разрядный IV.

На вход алгоритма поступают: *k*-разрядный ключ *K*, *m*-разрядный *IV*, последовательность *r*-разрядных блоков открытого текста $P = p_1, p_2, ..., p_u$.

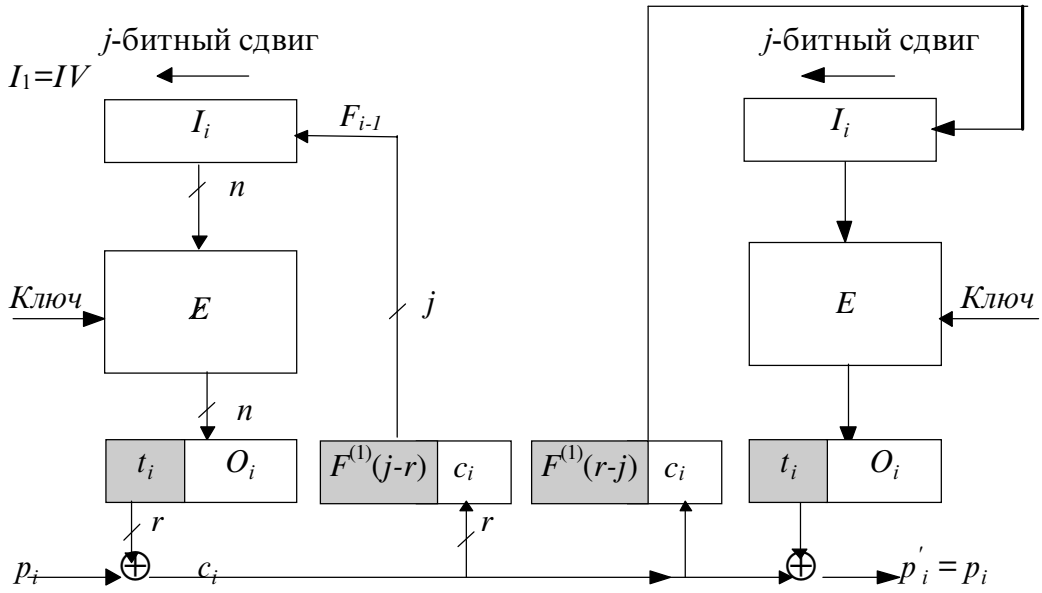


РИС. 9.11. *Режим CFB — r-битный открытый текст/j-битная обратная связь (r/j CFB режим).*

На выходе алгоритма формируется последовательность *r*-разрядных блоков закрытого текста $C = c_1, c_2, ..., c_u$.

Вначале осуществляется запись в буфер обратной связи (регистр сдвига I_i) значение вектора инициализации $I_1=IV$.

Далее для всех $i = \overline{1,u}$ выполнить следующие операции:

1. Выбрать *n* крайних слева бит из регистра сдвига *I*:

$$X_i = I_i \sim n;$$

2. Вычислить функцию шифрования *E*:

$$O_i = E_k(X_i);$$

3. Выборка *r* крайних слева бит из величины O_i:

$$t_i = O_i \sim r;$$

4. Сформировать и передать блок закрытого текста:

$$C_i = p_i \oplus t_i;$$

5. Сформировать переменную обратной связи из блока шифртекста c_i путем его дополнения слева *j-r* единицами:

$$F_i = F^{(1)}(j-r) \,||\, c_i;$$

6. Выполнить сдвиг содержимого буфера обратной связи *I* на *j* разрядов влево и записать F_i в регистр:

$$I_{i+1} = 2^j I_i + F_i \bmod 2^n.$$

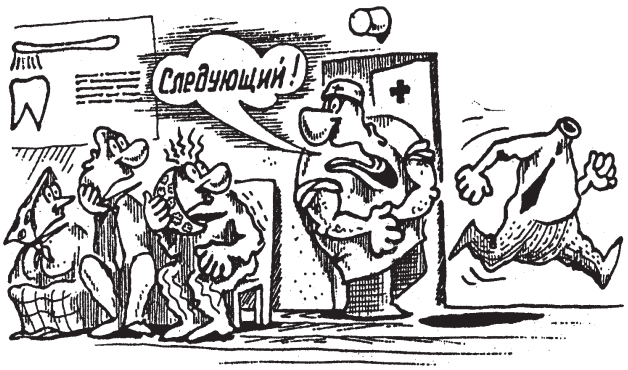
Дешифрование выполняется аналогичным образом, за исключением операции 4, а именно:

$$P_i = c_i \oplus t_i$$

На рис. 9.12 схематично представлен *CFB* режим по ISO/IEC 10116. Заметим также, что во второй редакции ISO/IEC 10116 рекомендуется выбирать *j = r*.

Режим обратной связи по выходу (034)

Режим *OFB* может быть использован для приложений в которых должно быть исключено любое размножение ошибок. Режим схож с режимом *CFB* и позволяет шифровать блоки различной длины, но в качестве об-



Режим обратной связи по выходу...

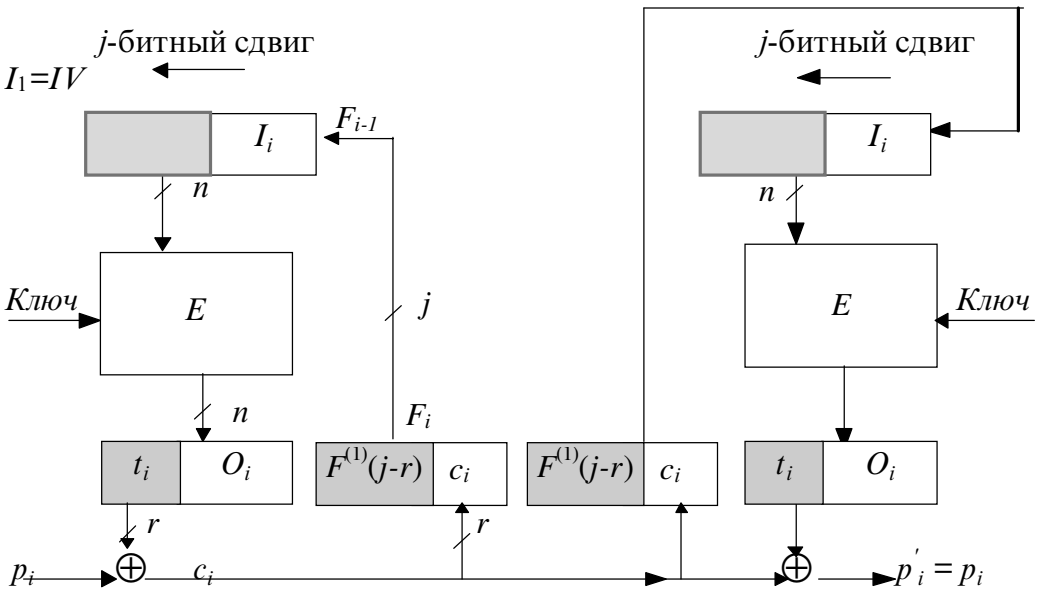


РИС. 9.12. *Режим CFB по ISO/IEC 10116 с конвейерной обработкой данных*

ратной связи используется не блок шифртекста, а шифрованный блок с выхода функции *E*.

Распространены две версии *OFB* режима работы *n*-разрядного блочного шифра. Версия ISO/IEC 10116: 1997 (ISO/IEC 10118: 1991) требует *n*-разрядной обратной связи (полная обратная связь) и является более стойкой. Ранее была принята версия FIPS 81, позволяющая $r \le n$ разрядную обратную связь.

На рис. 9.13. представлена схема *OFB* режима по ISO/IEC 10116: 1997.

В этой версии *OFB* режима на вход алгоритма поступают: *k*-битный ключ *K*, *n*-разрядный *IV*, последова-

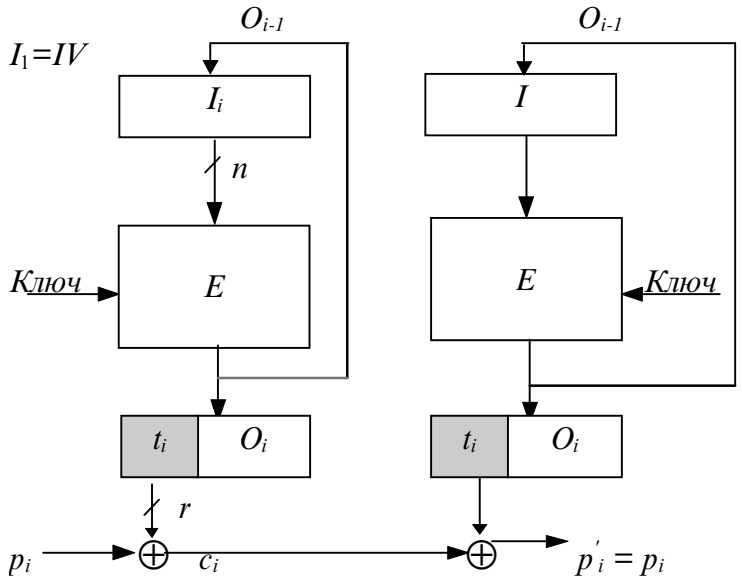


РИС. 9.13. *Режим OFB r-битный открытый текст/n-битная обратная связь (r/n — OFB режим)*

тельность r-разрядных блоков открытого текста $P = p_1, p_2,..., p_u$, $1 \le r \le n$. На выходе формируется последовательность r-разрядных блоков шифртекста.

Шифрование осуществляется следующим образом.
Сначала в регистр I_i записывается значение IV.
Затем для $1 \le i \le u$ выполнить:

1. Вычислить шифрованный блок:

$$O_i=E_k(I_i).$$

2. Выбрать r крайних слева бит величины O_i. Предполагают, что крайний бит определен как 1.

$$t_i=O_i^{\sim}r.$$

3. Передать r-разрядный блок шифртекста c_i:

$$c_i=p_i \oplus t_i.$$

4. Обновить содержимое регистра сдвига I_i:

$$I_{i+1}=O_i.$$

При дешифровании устанавливают $I_1=IV$ и для всех $1 \le i \le u$ по получении блока c_i осуществляют те же действия за исключением операции 3, которая имеет вид:

$$p_i=c_i \oplus t_i.$$

При реализации режима *OFB* и FIPS 81: k-битный ключ K, n-разрядный IV, последовательность r-разрядных блоков открытого текста $P = p_1, p_2,..., p_u$, где $1 \le r \le n$. На выходе формируется последовательность r-разрядных блоков шифртекста $C = c_1, c_2 ,..., c_u$.

Алгоритм шифрования и дешифрования аналогичен рассмотренным выше, за исключением того, что операция обновления регистра $I_{i+1} = O_i$ заменяется на операцию сдвига содержимого регистра и записи в него значения t_i:

$$I_{i+1}=2^rI_i + t_i \, mod 2^n.$$

Рассмотрим свойства OFB режима.

1. ***Шифрование идентичных открытых текстов***. Как и в *CBC* и в *CFB* режимах шифрования, изменение IV приводит к различным результатам шифрования одного и того же открытого текста.

2. ***Эфект сцепления блоков***. Ключевой поток является независимым от открытого текста.

3. ***Размножение ошибок при дешифровании.*** Одна или более битовых ошибок в любом символе шифртекста c_i, оказывает влияние на дешифрование только этого символа, и точно в тех же битовых позициях, в которых оказались ошибки в c_i. Таким образом обеспечивается полное восстановление бит открытого текста.

4. ***Восстановление синхронизации***. *OFB* режим восстанавливается после ошибок в битах шифртекста, но не может самосинхронизироваться после потери бит шифртекста, которые разрушают выравнивание (синхронизацию) дешифрующего ключевого потока (в этих случаях необходима пересинхронизация шифра).

5. ***Производительность***. Для $r < n$ производительность уменьшается в n/r раз (как и в *CFB* режиме). Однако, во всех случаях, поскольку ключевой поток не зависит от открытого и закрытого текста, он может быть вычислен заранее по заданному ключу и IV.

6. ***Изменение IV в OFB***. Вектор инициализации, который не является секретным, должен изменяться, если в *OFB* режиме повторно используется ключ K. В противном случае будет сформирован идентичный ключевой поток, и путем сложения по модулю 2 (*XOR*) соответствующих шифртекстов противник может прийти к криптоанализу, в котором шифр будет с бесконечным ключом, где в качестве ключа используется открытый текст.

Таким образом, исходя из свойств режима на практике режим *OFB* используется только для обеспечения функции конфиденциальности.

В упрощенном *OFB* режиме вводят обновление входного блока, как функцию счетчика, то есть:

$$I_{i+1}= I_i + 1.$$

Это позволяет избежать проблему так называеиого короткого цикла, и обеспечить восстанавлеваемость после ошибок в вычислении функции шифрования E. Более того, это обеспечивает свойство случайного доступа: не обязательно дешифровать i-й блок шифртекста для того, чтобы дешифровать i+1-ый блок.

Ранее мы упоминали, что версия ISO режима более стойкая, чем версия FIPS. В *OFB* режиме с полной n-разрядной обратной связью, ключевой поток генерируется с использованием итеративной функции $O_i=E_k(O_{i-1})$. Поскольку E_k является перестановкой и в предположении, что k является случайной величино, E_k действительно является случайным выбором из множества $(2^n)!$ перестановок по n элементам. Для фиксированных (случайных) значений ключа и вектора инициализации ожидаемая длина цикла перед повторением любого значения O_i будет равна 2^{n-1}. С другой стороны, если количество бит обратной связи равно $r < n$, как определено в FIPS 81, то ключевой поток будет формироваться с использованием итерации $O_i = f(O_{i-1})$, где f не является функцией перестановки. Предполагая ее поведение как случайной функции, ожидаемая длина цикла составит величину окло $2^{n/2}$. Следовательно предпочтительным является использование *OFB* режима с полной n-разрядной обратной связью.

Наконец необходимо отметить, что и *OFB* режим с полной обратной связью, и режим счетчика обеспечи-

вают возможность применение блочного шифра в качестве генератора ключевого потока для поточного шифра. Аналогичным образом и в *CFB* режиме осуществляется шифрование потока символов, при этом блочный шифр используется как генератор ключевого потока (зависимого от открытого текста). Режим *CBC* также можно рассматривать как поточный шифр с *n*-разрядными блоками, играющими роль очень больших символов. Таким образом, режимы работы блочных шифров позволяют построить

Резюме

Служба защиты — совокупность механизмов, процедур и других управляющих воздействий, реализованных для сокращения риска, связанного с угрозой. Например, службы идентификации и аутентификации (опознания) помогают сократить риск угрозы неавторизованного пользователя. Некоторые службы обеспечивают защиту от угроз, в то время как другие службы обеспечивают обнаружение реализации угрозы.

Примеры некоторых служб защиты:

- идентификация и установление подлинности — является службой безопасности, которая помогает гарантировать, что в ИС работают только авторизованные лица.
- управление доступом — является службой безопасности, которая помогает гарантировать, что ресурсы ИС используются разрешенным способом.
- конфиденциальность данных и сообщений — является службой безопасности, которая помогает гарантировать, что данные ИС, программное обеспечение и сообщения не раскрыты неавторизованным лицам.
- целостность данных и сообщений — является службой безопасности, которая помогает гарантировать, что данные ИС, программное обеспечение и сообщения не изменены неправомочными лицами.
- контроль участников взаимодействия — является службой безопасности, посредством которой гарантируется, что объекты, участвующие во взаимодействии, не смогут отказаться от участия в нем. В частности, отправитель не сможет отрицать посылку сообщения (контроль участников взаимодействия с подтверждением отправителя) или получатель не сможет отрицать получение сообщения (контроль участников взаимодействия с подтверждением получателя).
- регистрация и наблюдение — является службой безопасности, с помощью которой может быть прослежено использование всех ресурсов ИС.

Приведенные службы защиты должны рассматриваться как возможные, а не обязательные решения.

Под ***аутентификацией*** пользователя (субъекта) понимается установление его подлинности.

Под ***идентификацией*** — определение тождественности пользователя или пользовательского процесса, необходимое для управления доступом. После идентификации обычно производится аутентификация.

Под ***авторизацией*** (санкционированием) подразумевается предоставление разрешения доступа к ресурсу системы.

В большинстве ИС используется механизм идентификации и аутентификации на основе схемы идентификатор пользователя/пароль. Аутентификация, которая полагается исключительно на пароли, часто не может обеспечить адекватную защиту. Пользователи имеют тенденцию создавать пароли, которые являются легкими для запоминания и, следовательно, легкими для угадывания. С другой стороны, если пользователи должны использовать пароли, сгенерированные из случайных символов, которые трудно угадывать, то пользователям также трудно их запомнить.

Управление доступом может быть достигнуто при использовании дискреционного управления доступом или мандатного управления доступом.

Дискреционное управление доступом — наиболее общий тип управления доступом, используемого в ИС. Основной принцип этого вида защиты состоит в том, что индивидуальный пользователь или программа, работающая от имени пользователя, имеет возможность явно определить типы доступа, которые могут осуществить другие пользователи (или программы, выполняющиеся от их имени) к информации, находящейся в ведении данного пользователя. Дискреционное управление доступом отличается от мандатной защиты, в том, что оно реализует решения по управлению доступом, принятые пользователем.

Мандатное управление доступом реализуется на основе результатов сравнения уровня допуска пользователя и степени конфиденциальности информации.

Существуют механизмы управления доступом, которые поддерживают степень детализации управления доступом на уровне следующих категорий:

- владелец информации,
- заданная группа пользователей
- все другие авторизованные пользователи.

Это позволяет владельцу файла (или каталога) иметь права доступа, отличающиеся от прав всех других пользователей и определять особые права доступа для указанной группы людей или всех остальных пользователей.

К сожалению, сегодня типично скорей интуитивное проявление интереса, например, к способам обезопасить свои данные, а не целенаправленная работа по

анализу уровня безопасности своих информационных систем. Это результат того, что лишь крупные корпоративные заказчики могут проводить дорогостоящие проекты по защите информации и содержать штат сотрудников, занимающихся развитием и поддержкой защитных систем. Поэтому очень важно понимание заказчиком своих потребностей по защите информации, проблем организации работ и своих финансовых возможностей.

Выделяются следующие группы средств защиты информации:

- средства защиты от НСД;
- системы анализа и моделирования информационных потоков (CASE-системы);
- системы мониторинга сетей;
- анализаторы протоколов;
- антивирусные средства;
- межсетевые экраны;
- криптографические средства;
- системы резервного копирования;
- системы бесперебойного питания;
- системы аутентификации;
- средства предотвращения взлома корпусов и краж оборудования;
- средства контроля доступа в помещения;
- инструментальные средства анализа системы защиты.

Криптографические хеш-функции играют фундаментальную роль в современной криптографии. Особенно широко они используются при обеспечении целостности данных и аутентификации сообщений. Аутентичность сообщения можно обеспечить различными способами, не прибегая к его шифрованию. Такой подход пригоден во многих случаях, когда целостность и аутентичность данных играет исключительно важную роль, а конфиденциальность не требуется, например, при реализации финансовых операций и распределении открытых ключей между объектами. ***Широко распространены следующие методы обеспечения подлинности сообщения:***

- добавление к сообщению кода подлинности сообщения (код аутентификации сообщения) (message authentification code, MAC-код) или зашифрованной контрольной суммы;
- введение цифровых подписей.

Специализированные хеш-функции (Customized hash functions или dedicated hash functions) специально разработаны только для целей хеширования и оптимизированы для выполнения данной задачи. Третья часть стандарта ISO/IEC 10118-3 определяет три специализированные хеш-функции, а именно функции RIPEMD-128, RIPEMD-160 и SHA-1. Данные хеш-функции основаны на использовании принципов построения, заложенных в хеш-функции семейства MD*x* (MD2, MD4,MD5), которые специально разрабатывались для реализации на 32-разрядных ЭВМ. Алгоритм MD4 был предложен Р. Райвестом в 1990 году, а в 1991 тот же автор предложил модифицированную версию алгоритма MD5. В настоящее время хеш-функции MD4, MD5 являются наиболее распространенными в практических приложениях хеш-функциями. Однако некоторые их недостатки не позволили стандартизировать их.

Для описания процессов обработки информации с использованием механизмов ЦП воспользуемся следующей терминологией.

1. ***Алгоритм генерации ЦП*** – это метод формирования ЦП.

2. ***Алгоритм проверки (верификации) ЦП*** – метод проверки того, что подпись является аутентичной, т.е. действительно создана конкретным объектом и не модифицирована при передаче.

3. ***Схема ЦП*** **(или** ***механизм ЦП*****)** – совокупность взаимосвязанных алгоритмов генерации и верификации цифровой подписи.

4. ***Процесс*** **(*процедура*)** ***наложения ЦП*** – совокупность математического алгоритма генерации ЦП и методов представления (форматирования) подписываемых данных.

5. ***Процесс*** **(*процедура*)** ***снятия ЦП*** – совокупность алгоритма верификации ЦП и методов восстановления данных.

Защита информации — не разовое мероприятие и даже не совокупность мероприятий, а непрерывный процесс, который должен протекать (осуществляться) во все время и на всех этапах жизненного цикла ИС;

Осуществление непрерывного процесса защиты информации возможно лишь на базе промышленного производства средств защиты.

Поддержание и обеспечение надежного функционирования механизмов защиты информации в ИС сопряжено с решением специфических задач и поэтому может осуществляться лишь профессионально подготовленными специалистами.

Защита информации к настоящему времени уже выросла в достаточно серьезное и относительно самостоятельное научное направление, составляющее одну из ветвей фундаментального научного направления информатики.