Основы информационной безопасности

проф. А.В. Цыганов, СПбГУ, 2008

Под ***информационной безопасностью*** понимается защищенность информации и поддерживающей ее инфраструктуры от любых случайных или злонамеренных воздействий, результатом которых может явиться нанесение ущерба самой информации, ее владельцам или поддерживающей инфраструктуре.

Задачи информационной безопасности сводятся к минимизации ущерба, а также к прогнозированию и предотвращению таких воздействий.

Информация должна быть

✓сохранена от неправомочного доступа (**конфиденциальность**),

✓ защищена от неправомочного изменения (**целостность**)

✓доступна только разрешенному объекту, когда это ему необходимо (**готовность**)

В англо-американской традиции различают два основных вида **конфидециальности**:

- добровольную (**privacy**)
- принудительную (**secrecy**)

В первом случае имеются в виду прерогативы личности, во втором случае имеется в виду информация для служебного пользования, доступная ограниченному кругу официальных лиц.

Хотя privacy и secrecy схожи по значению, ***на практике они обычно противоречат друг другу: усиление secrecy ведёт к нарушению и уменьшению privacy***.

Целостность означает, что изменения должны быть сделаны только **разрешенными объектами** и с помощью **разрешенных механизмов**.

Нарушение целостности — не обязательно результат злонамеренного действия; сбой в системе, такой, например, как всплеск или прерывание мощности в первичной сети электропитания, может привести к нежелательным изменениям некоторой информации.

Целостности данных можно угрожать несколькими видами атак, такими как **модификация, имитация источника, повторная передача информации и отказ от сообщения.**

Третий компонент информационной безопасности — **готовность**.

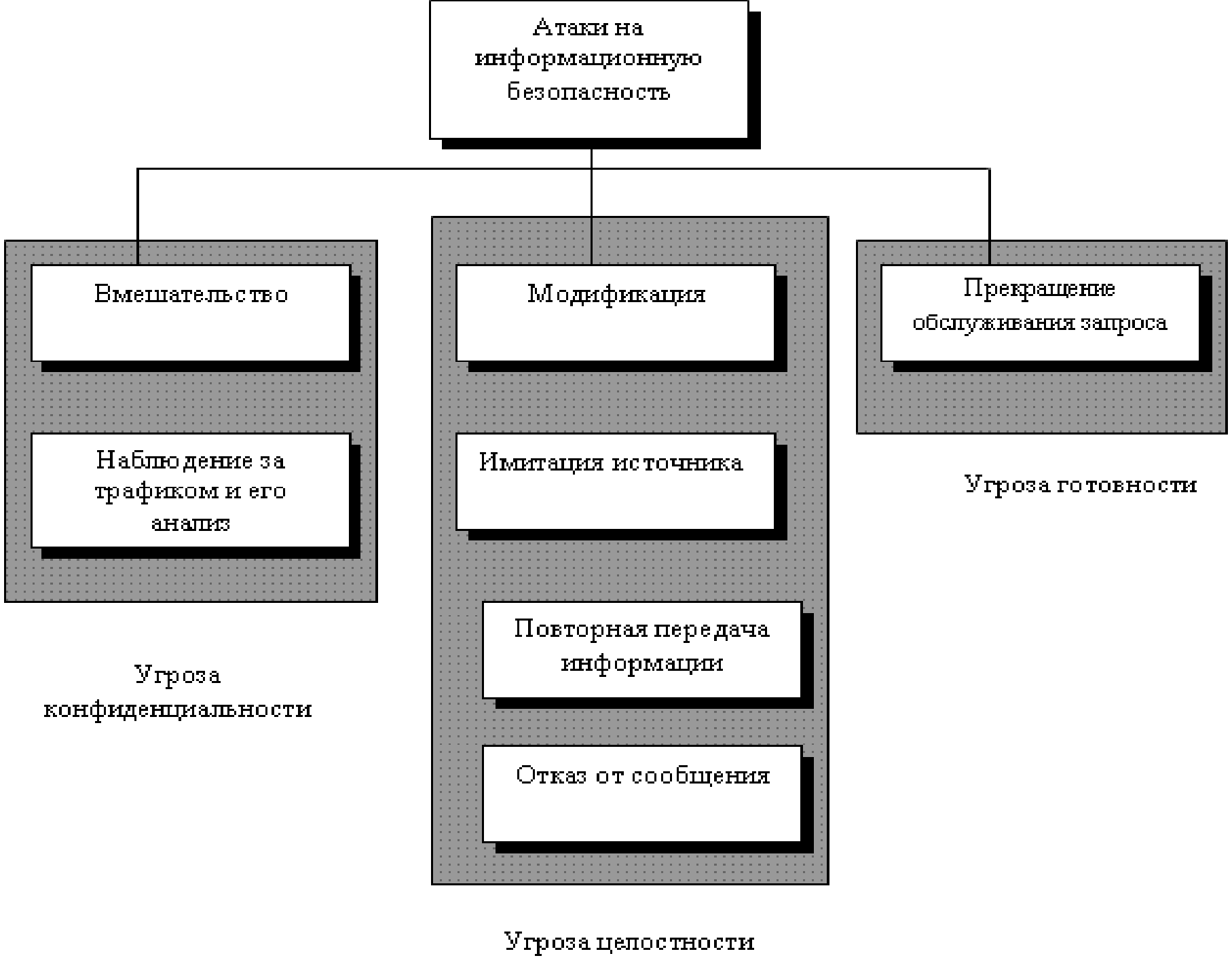
Информация, созданная и сохраненная организацией, должна быть доступна разрешенным объектам.

Информация бесполезна, если она не доступна.

Информация должна постоянно изменяться, и поэтому тоже должна быть доступна для разрешенных объектов.

Неготовность информации столь же вредна для организации, как отсутствие конфиденциальности или целостности.

Вообразите, что случилось бы с банком, если клиенты не могли бы обратиться к своим счетам для снятия или вклада денег.


Атаки на информационную безопасность
Вмешательство
Наблюдение за трафиком и его анализ
Угроза конфиденциальности
Модификация
Имитация источника
Повторная передача информации
Отказ от сообщения
Угроза целостности
Прекращение обслуживания запроса
Угроза готовности

Стеганография и криптография

- **Стеганография** **и** **криптография** и есть лишь очень маленькие (но важные) части системы обеспечения информационной безопасности.

- Большинство систем взламываются за счет слабостей других частей, например
  - неправильных требований или спецификаций;
  - ошибок реализации;
  - уязвимостей на уровне приложений;
  - человеческого фактора (social engineering).

Стеганография (steganography)

- буквально «тайнопись» - наука о скрытии наличия информации с использованием технических, компьютерных и математических методов.

В отличие от криптографии стеганография скрывает не только информацию, но и сам факт её наличия.

Ещё Геродот описывал послания, написанные на деревянных дощечках.

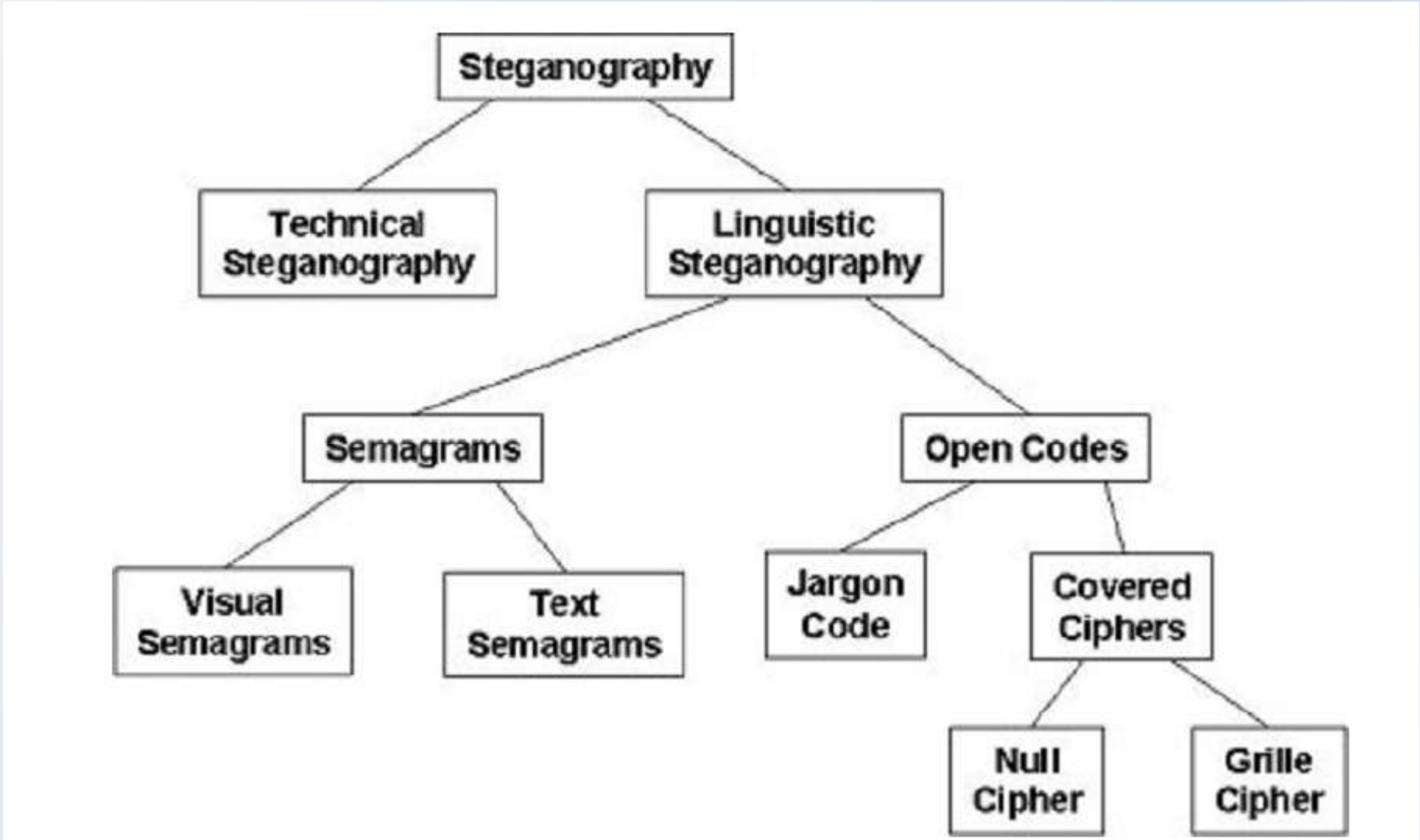
В отличие от обычного способа записи, когда сначала наносился слой воска, а потом писался текст, здесь секретная запись выцарапывалась прямо на дощечке, которую потом покрывали воском, где уже и писали - чаще всего, ложное сообщение.

Т.е. одним из наиболее распространенных и старых методов **классической стеганографии** является использование симпатических (невидимых) чернил..

Стеганография скрывает тайное сообщение, но не факт того, что две стороны общаются друг с другом. Процесс стеганографии, как правило, включает в себя размещение скрытого сообщения в некотором носителе для транспортировки, который называется контейнер.
Естественно, такая операция должна остаться незамеченной - файл-контейнер обязан не терять функций, а наличие скрытого сообщения должно быть максимально сложно обнаружить.

Примеры — стеганографическая файловая система StegFS для Linux, скрытие данных в неиспользуемых областях форматов файлов, подмена символов в названиях файлов, текстовая стеганография и т. д.

Таксономия стеганографии:



· ***Техническая стеганография*** (technical steganography) использует научные методы для скрытия сообщения, такие как, использования невидимых чернил или микрофотоснимков и другие методы сокращения размера;

· ***Лингвистическая стеганография*** (linguistic steganography) скрывает сообщение в контейнере некоторыми неочевидными способами и далее классифицируется как семаграммы (semagrams) или открытые коды (open codes);

· ***Семаграммы*** скрывают информацию при помощи символов или знаков.

Визуальная семаграмма (visual semagrams) использует безобидные на первый взгляд или обычные физические объекты для передачи сообщения, например, каракули или расположения элементов на рабочем столе или веб-сайте.

Текстовая семаграмма (text semagrams) скрывает сообщение, изменяя внешний вид текста-контейнера, например, едва различимые изменения в размере или типе шрифта, добавляя дополнительные пробелы или различные завитушки в буквах или рукописном тексте;

· ***Открытые коды*** скрывают сообщение в «законном» сообщении-контейнере такими способами, которые не видимы для неподозревающего наблюдателя. Эта категория подразделена на жаргонные коды и скрытые шифры;

· ***Жаргонный код*** - используем язык, который понятен одной группе людей, но не имеет смысла для других. Жаргонные коды включают в себя нанесение пиктограмм, тайную терминологию, или невинный разговор, который передает особый смысл вследствие того, что факты известны только говорящим.

Подкласс жаргонных кодов – коды условных знаков, когда значение передают некие заранее подготовленные фразы.

Скрытые или замаскированные шифры (covered ciphers) скрывают сообщение в носителе-контейнере так, чтобы его мог восстановить любой, кто знает секрет того, как оно было скрыто.

Шифр “ решетка” (grille cipher) применяет шаблон, который используется, чтобы скрыть сообщение-контейнер. Слова, которые появляются в отверстиях шаблона, являются скрытым сообщением.

Нулевой шифр (null cipher) скрывает сообщение согласно некоторому заранее подготовленному набору правил, например, «прочитайте каждое пятое слово» или «посмотрите на третью букву в каждом слове».

В мире компьютерных технологий ***стеганография*** предоставляет некоторые очень полезные и коммерчески важные функции, наиболее известные из которых – создание цифровых водяных знаков.

В этом приложении, автор может вставить скрытое сообщение в файл, чтобы позднее можно было доказать право интеллектуальной собственности и/или гарантировать целостность содержимого.

Художник, например, может поместить оригинал иллюстрации на веб-сайте. Если кто-нибудь другой украдёт файл и заявит, что эта работа является его собственностью, художник может позже доказать право собственности, потому что только он или она может восстановить водяной знак.

***Естественно, что у стеганографии есть ряд совершенно* незаконных приложений.**

Наиболее известны те, которые скрывают записи о незаконной деятельности, финансовом мошенничестве, индустриальном шпионаже и обмен информацией между членами преступных или террористических организаций и т.д. ☺

Типы компьютерной стеганографии

1. Методы, использующие особенности компьютерных форматов.

Конкретные примеры - поле комментариев в формате JPEG и поле Company в свойствах исполняемых EXE.

2. Алгоритмы, использующие избыточность аудиовизуальной информации.

Второе название этого метода - метод младших бит. Основными контейнерами в данном способе скрытия являются форматы так называемого прямого кодирования, например, BMP для графики, или WAV для звука.

3. Скрытие сообщений на цифровых носителях.

Например, данные можно размещать в зазорах файловой системы или свободной области, как остатки предыдущих файлов, и можно написать программы для получения доступа непосредственно к зазорам и свободным областям файловой системы. Небольшие количества данных также могут быть скрыты в неиспользуемых частях заголовков файлов.

4. Сетевые протоколы могут быть ещё одним цифровым контейнером. Например тайный протокол управления передачей Роуланда формирует скрытые каналы связи, используя поле идентификации в пакетах протокола IP или поле порядкового номера в сегментах протокола управления передачей (TCP) .

5. Существует также несколько **звуковых характеристик**, которые могут быть изменены таким образом, что будут неразличимыми человеческими чувствами, и такие изменения, например, небольшие изменения фазового угла, модуляции речи и частоты, могут переносить скрытую информацию**.**

Из-за огромного количества различных файлов-контейнеров, и свободного доступа к соответствующему стеганографическому программному обеспечению, которое будет работать с этими контейнерами, графические и аудио файлы остаются самыми легкими и распространёнными носителями-контейнерами в Интернете.

Стеганографию не найдут, только если её не будут искать.

Существует несколько отчётов о том, что террористы Аль-Каиды использовали порнографические графические файлы в качестве стеганографических носителей ., которые даже не проверялись соответствующими лицами на наличие скрытых сообщений.

Использование данным конкретным противником стеганографии и порнографии было очень неожиданным с технологической и культурной точки зрения, но это демонстрирует необходимость развивать в себе способность работать по-новому.

Литература:

Конахович Г. Ф., Пузыренко А. Ю. ***Компьютерная стеганография. Теория и практика*****. — К.: «МК-Пресс», 2006.**

Грибунин В. Г., Оков И. Н., Туринцев И. В. ***Цифровая стеганография*****. — М.: «Солон-Пресс», 2002.**

http://computer-forensics-lab.org/lib/data/129.pdf

Криптография

Криптография - ***наука о защите информации с использованием математических методов***.

Криптоанализ - наука противоположная криптографии и посвященная методам вскрытия защищенной информации.

Совокупность криптографии и криптоанализа принято называть криптологией.

Криптология

(греч.: скрытое слово), наука о математических аспектах защиты информации

Криптография,

наука о шифровании, алгоритмах кодирования данных

Криптоанализ,

наука о "взломе" шифров

Что такое криптография?

- Наука о том
  - как сделать информацию конфиденциальной, избирательно доступной (шифрование)
  - как обеспечить целостность данных
  - как обеспечить аутентификацию (достоверную идентификацию)
    - **субъекта**: аутентичность информационного источника
    - **объекта**: пользователя, процесса
  - как обеспечить доказательность действия (неотказуемость)
  - как обеспечить контроль доступа (авторизацию)

Предмет науки:

криптографические алгоритмы (математика)

криптографические протоколы (процессы с использованием криптографических алгоритмов)

Принцип (Август Керхоффс, 1835-1903):

вся защита должна основываться *только* на качестве (длине, энтропии) **ключа**

алгоритмы должны быть тщательно выверены и **публично доступны**

Метод:

для того, чтобы выполнить криптографическую операцию (за исключением, м.б., обеспечения целостности данных), нужно знать секретную информацию (**ключ**) незнающий ключа должен «искать иголку в стоге сена» (а «стог» должен быть достаточно большим в математическим смысле)

Основные проблемы криптографии

- Обеспечение долговременной стойкости.
- Создание криптографических схем с низкой ресурсоемкостью (low footprint) и небольшим потреблением энергии.
- Обеспечение высокой производительности.
- Разработка легко переносимых алгоритмов (algorithm agility).
- Реализация криптографических схем, не снижающая их стойкости.

Криптографические методы могут быть классифицированы различным образом, но наиболее часто они подразделяются в зависимости от количества ключей:

Бесключевые, в которых не используются какие-либо ключи.

Одноключевые - в них используется некий дополнительный ключевой параметр - обычно это секретный ключ.

Двухключевые, использующие в своих вычислениях два ключа: секретный и открытый.


Криптографические методы
Бесключевые
Одноключевые
Двухключевые
Методы криптографического контрольного суммирования
Генерация случайных чисел
Симметричное шифрование
Методы криптографического контрольного суммирования
Генерация псевдослучайных чисел
Аутентификация
Асимметричное шифрование
Электронная подпись
Аутентификация

1. **Электронная подпись** используется для подтверждения целостности и авторства данных.

Целостность данных означает, что данные не были случайно или преднамеренно изменены при их хранении или передаче.

Алгоритмы электронной подписи используют два вида ключей:

- **секретный ключ** используется для вычисления электронной подписи;
- **открытый ключ** используется для ее проверки.

При использовании криптографически сильного алгоритма электронной подписи и при грамотном хранении и использовании секретного ключа (то есть при невозможности использования ключа никем, кроме его владельца) никто другой не в состоянии вычислить верную электронную подпись какого-либо электронного документа.

Электронная подпись защищает документы от следующих угроз:

- ✓подготовка документа от имени другого субъекта ("маскарад"),
- ✓отказ автора документа от авторства (ренегатство),
- ✓изменение содержания документа получателем (подмена),
- ✓изменение содержания третьим лицом (перехват),
- ✓повторная передача уже переданного документа (повтор).

Автор шифрует документ с помощью секретного ключа и передает его вместе с другим ключом - открытым.

Windows имеет набор функций CryptoAPI для выполнения операций шифрования, расшифрования, получения и проверки ЭП, генерации, хранения и распределения ключей шифрования

2. **Аутентификация** позволяет проверить, что пользователь (или удаленный компьютер) действительно является тем, за кого он себя выдает.

Простейшей схемой аутентификации является парольная - в качестве секретного элемента в ней используется пароль, который предъявляется пользователем при его проверке.

Такая схема доказано является слабой, если для ее усиления не применяются специальные административно-технические меры.

На на основе **шифрования** или **хэширования** можно построить действительно сильные схемы аутентификации пользователей.

3. **Криптографическое контрольное суммирование**:

- ключевое и бесключевое хэширование;
- вычисление имитовставок;
- использование кодов аутентификации сообщений.

Фактически, все эти методы различным образом из данных произвольного размера с использованием секретного ключа или без него вычисляют некую контрольную сумму фиксированного размера, **однозначно** соответствующую исходным данным.

Криптографическое **контрольное суммирование** широко используется в различных методах защиты, анализа и обработки информации.

- для подтверждения целостности любых данных в тех случаях, когда использование электронной подписи невозможно (например, из-за большой ресурсоемкости) или является избыточным;

- в самих схемах электронной подписи - "подписывается" обычно хэш данных, а не все данные целиком;

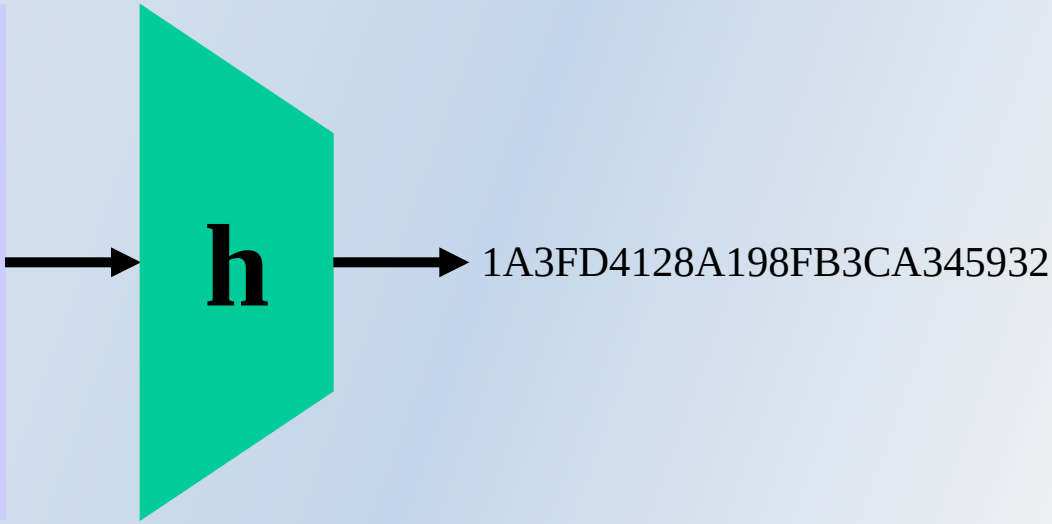
- в различных схемах аутентификации пользователей.

Хэш-функции

- Код обнаружения модификаций (manipulation detection code, MDC).
- Достаточно защитить короткое хэш-значение, а не длинный текст.

- трудность обнаружения (экзистенциальных) коллизий (collision resistance);
- трудность инвертирования (preimage resistance);
- трудность обнаружения специфических коллизий (2nd preimage resistance).

Это входное значение криптографической хэш-функции. Оно является очень длинной строкой, которая преобразуется хэш-функцией в строку фиксированной длины. При этом накладываются дополнительные требования к стойкости: должно быть очень трудно найти входное значение, отображающееся в данное хэш-значение (прообраз) или найти два различных входных значения , отображающиеся в одно и то же хэш-значение (коллизию).



4. **Генераторы случайных и псевдослучайных чисел** позволяют создавать последовательности случайных чисел, которые широко используются в криптографии, в частности:

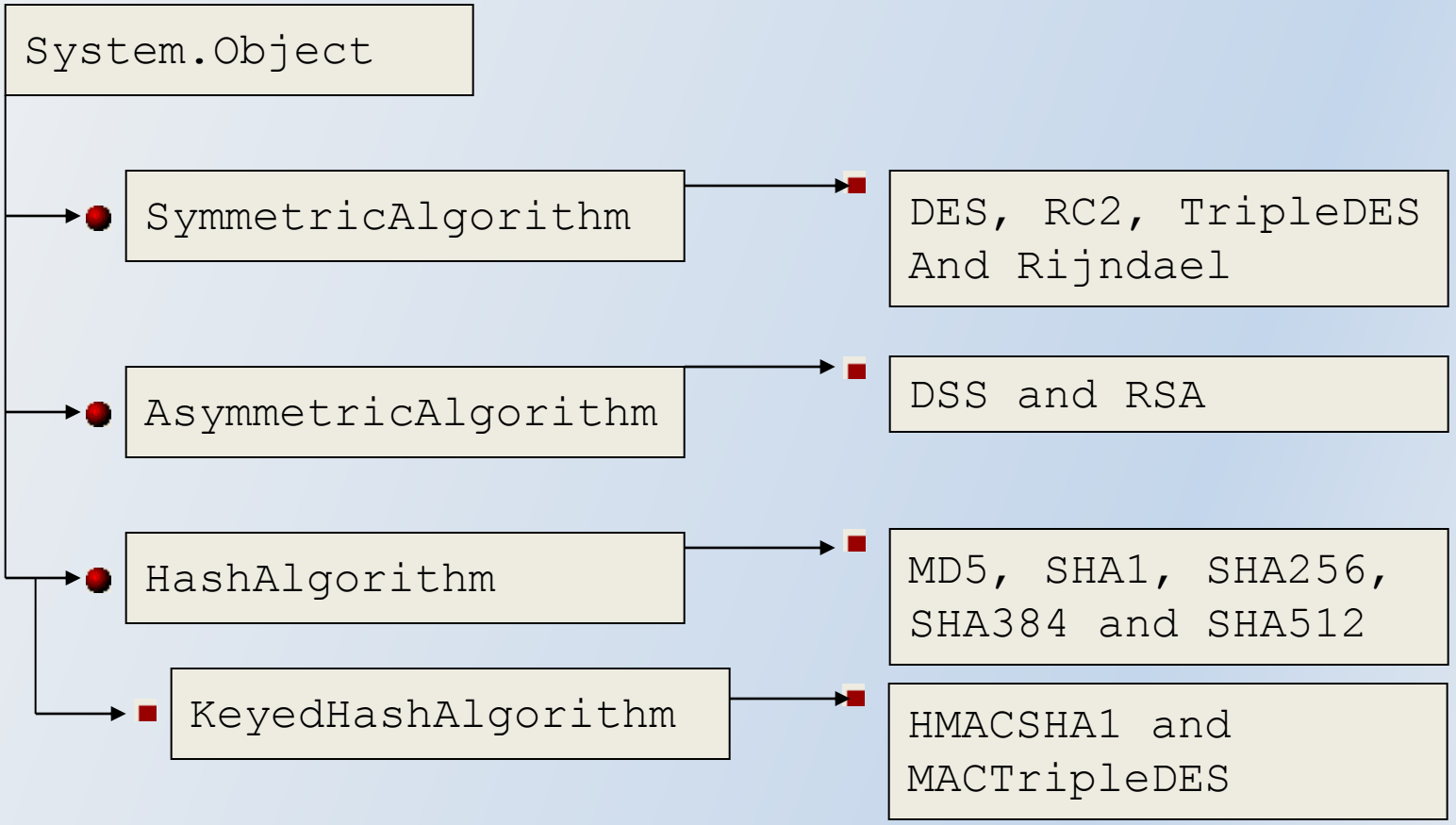
- случайные числа необходимы для генерации секретных ключей, которые, в идеале, должны быть абсолютно случайными;
- случайные числа применяются во многих алгоритмах электронной подписи;
- случайные числа используются во многих схемах аутентификации.

5. Шифрование информации - это преобразование открытой информации ***M*** (message) в зашифрованную ***C*** (которая чаще всего называется ***шифртекстом*** или ***криптограммой***), и наоборот.

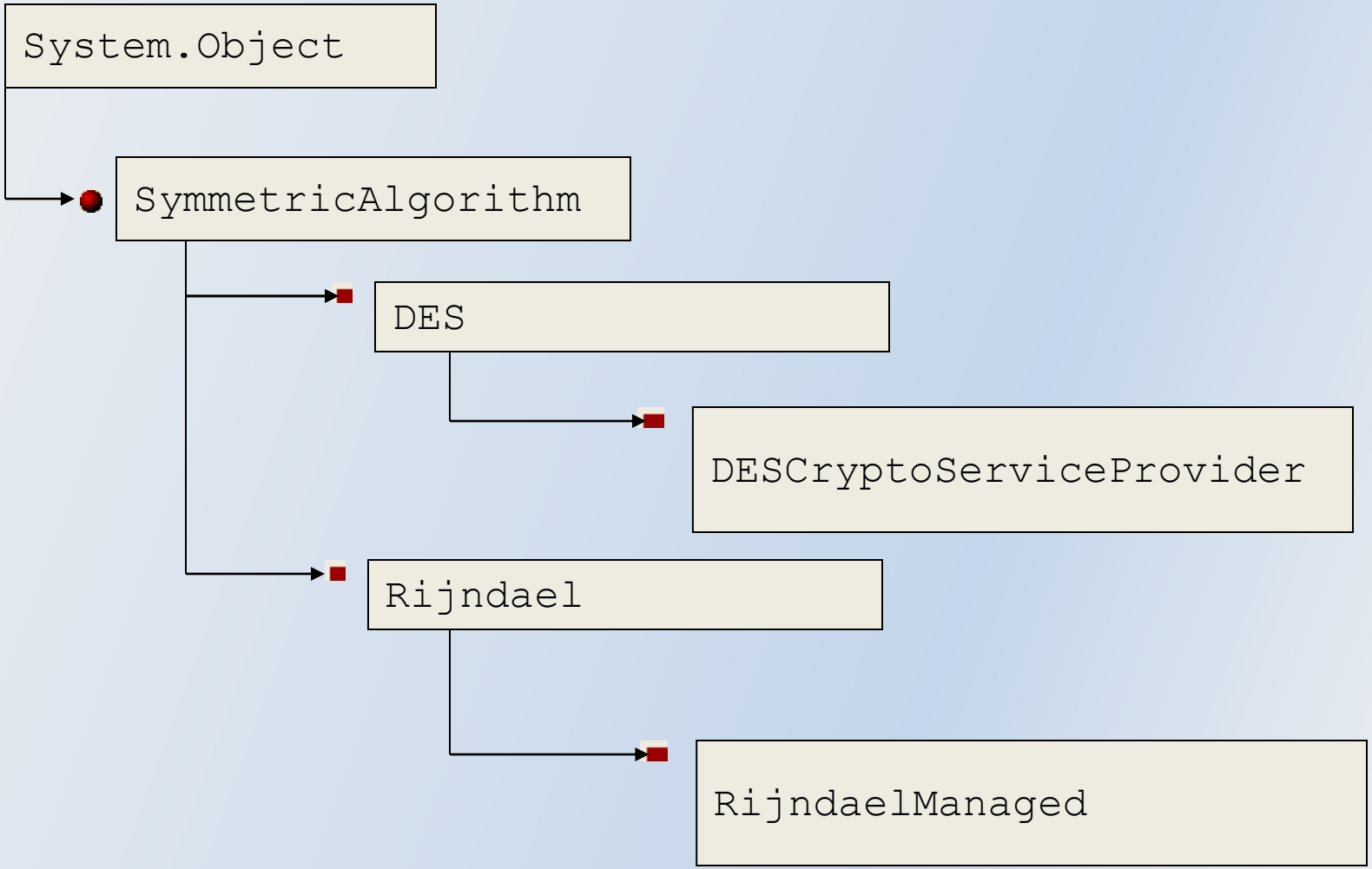
Первая часть этого процесса называется ***шифрованием***, вторая - ***расшифрованием***.

Большинство алгоритмов уже реализовано, например в .NET можно использовать следующие классы

System.Security.Cryptography



Структура классов



Информация дискретна – т.е.сообщение, которое должно быть зашифровано, состоит из последовательных дискретных символов, каждый из которых выбран из некоторого конечного множества (алфавита).

Эти символы могут быть буквами или словами некоторого языка, амплитудными уровнями "квантованной" речи или видеосигнала и т.д., но главный акцент будет сделан на случае **букв**.

Потоковые шифры

- **Исторически очень важны** (ввиду малых размеров).
  - Для A5/1 и E0, основанных на регистрах сдвига с линейной обратной связью, известны практически осуществимые атаки.
  - RC4, ориентированный на программную реализацию, имеет серьезные слабости.
  - Блоковый шифр в счетчиковом режиме или с зацеплением по выходу (более медленный).
- **В настоящее время**:
  - много взломанных схем;
  - исключения: SNOW2.0, MUGI;
  - недостаток стандартов и решений, доступных в открытой литературе.

Секретная система:

определяется абстрактно как некоторое множество отображений одного пространства (множества возможных сообщений ***M***) в другое пространство (множество возможных криптограмм ***C***).

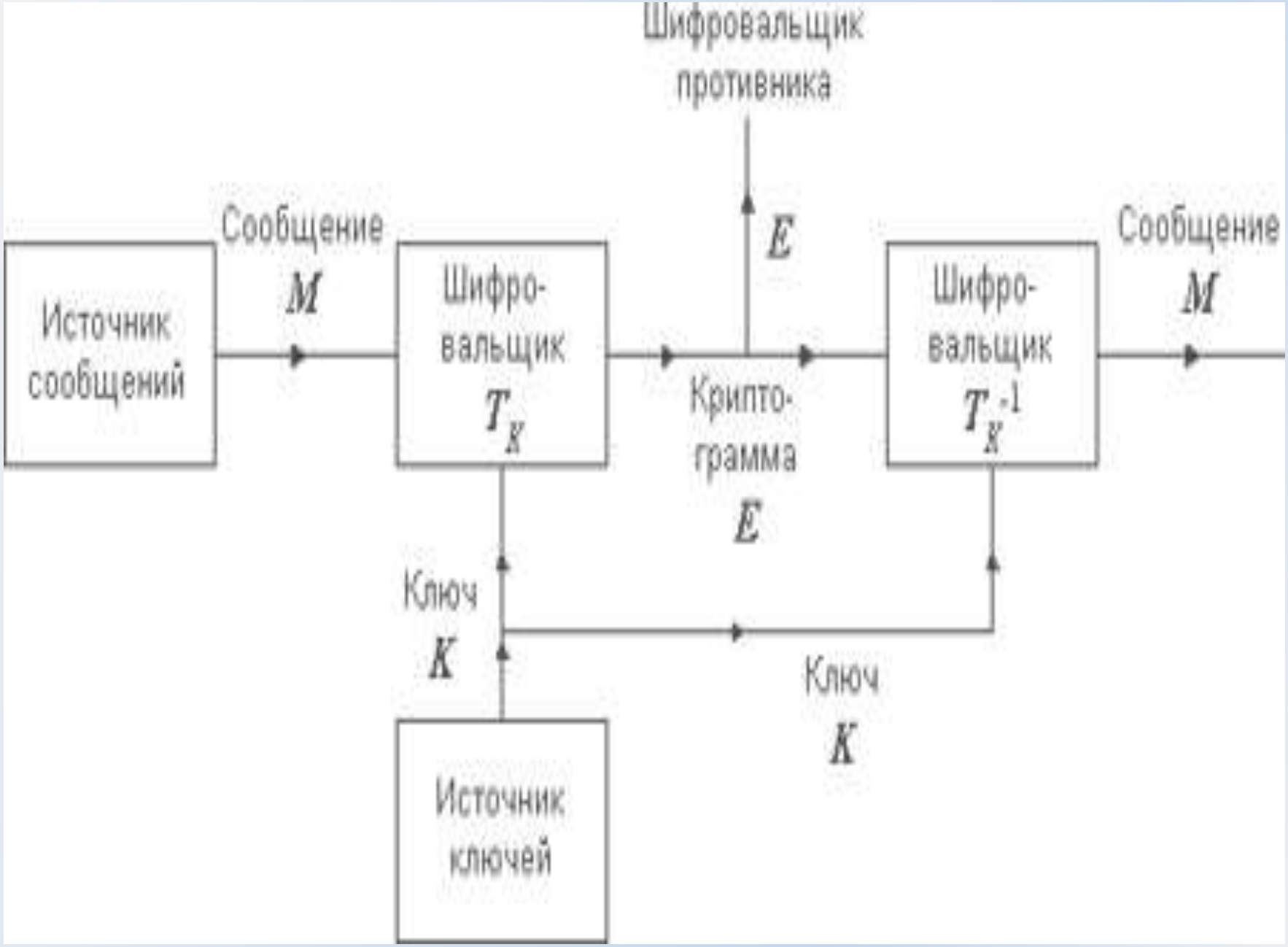
Каждое конкретное отображение из этого множества соответствует способу шифрования при помощи конкретного ключа.

Предполагается, что отображения являются ***взаимнооднозначными***, так что если известен ключ, то в результате процесса расшифрования возможен лишь единственный ответ.

Предполагается, что каждому **ключу** (и, следовательно, каждому отображению) соответствует некоторая априорная **вероятность** - вероятность выбрать этот ключ.

Аналогично каждому возможному сообщению ***M*** соответствует априорная вероятность, определяемая задающим сообщение вероятностным процессом.

Эти вероятности различных ключей и сообщений являются фактически априорными вероятностями для шифровальщика противника и характеризуют его априорные знания относительно интересующей его проблемы.


Шифровальщик
противника
Сообщение
M
Источник
сообщений
Шифро-
вальщик
T_K
E
Крипто-
грамма
E
Шифро-
вальщик
T_K^{-1}
Сообщение
M
Ключ
K
Ключ
K
Источник
ключей

Оценка секретных систем.

- Количество секретности.
- Объем ключа.
- Сложность операции зашифрования и расшифрования.
- Разрастание числа ошибок.
- Увеличение объема сообщения.

Основной характеристикой алгоритма шифрования является ***криптостойкость***, которая определяет его стойкость к раскрытию методами ***криптоанализа***.

Обычно эта характеристика определяется **интервалом времени**, необходимым для раскрытия шифра и количеством привлеченных ресурсов.

Симметричное шифрование

Симметричное шифрование менее удобно из-за того, что при передаче зашифрованной информации кому-либо необходимо, чтобы адресат заранее получил ключ для расшифрования информации.

У ***асимметричного шифрования*** такой проблемы нет (поскольку открытый ключ можно свободно передавать по сети), однако, есть свои проблемы, в частности, проблема подмены открытого ключа и медленная скорость шифрования.

Наиболее часто асимметричное шифрование используется в паре с симметричным - для передачи ключа симметричного шифрования, на котором шифруется основной объем данных.

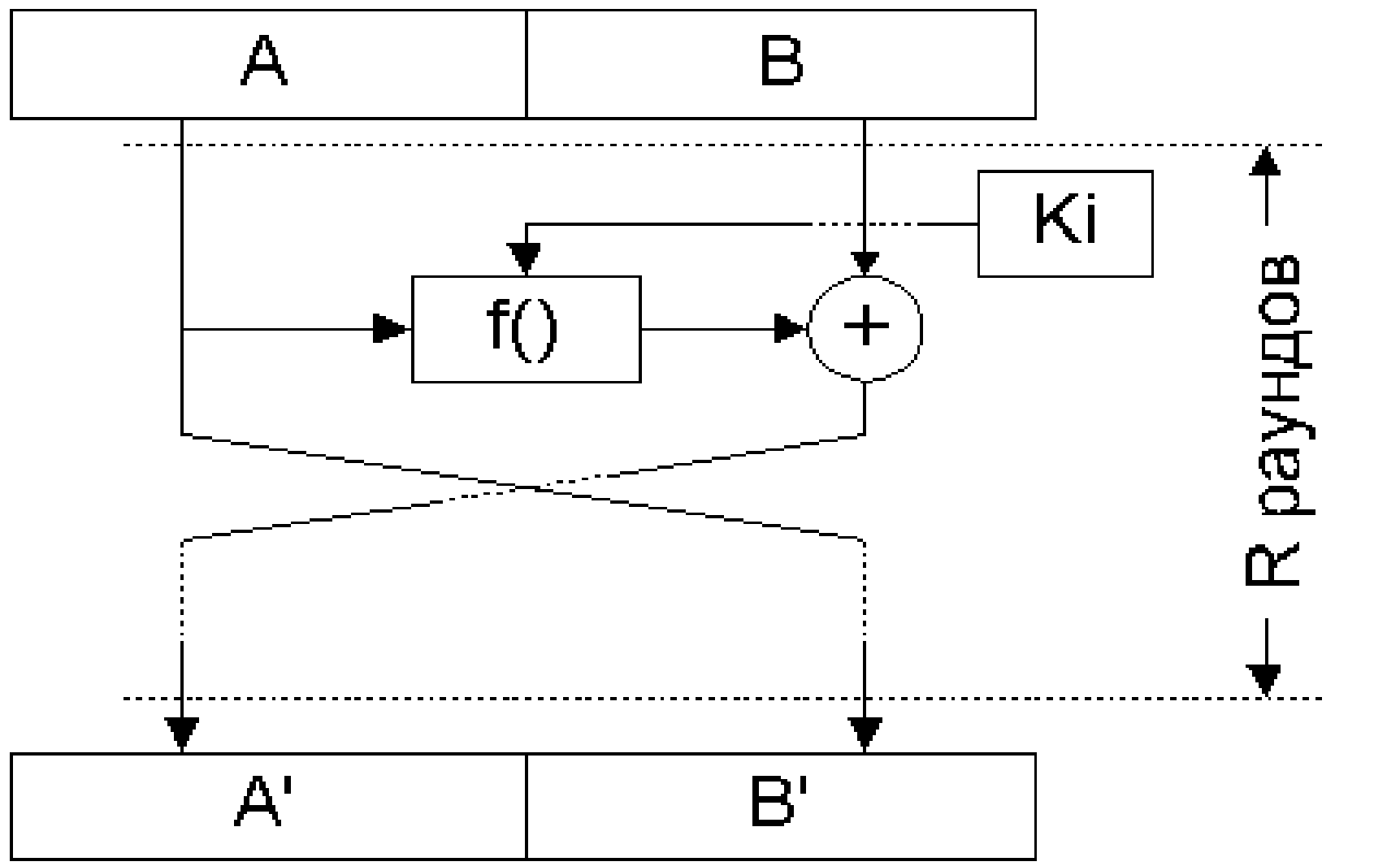
В настоящее время симметричное шифрование используется гораздо чаще асимметричного, поэтому остальная часть лекции будет посвящена только симметричному шифрованию.

Симметричное шифрование бывает двух видов:

1. ***Блочное шифрование*** - информация разбивается на блоки фиксированной длины (например, 64 или 128 бит), после чего эти блоки поочередно шифруются. Блоки могут шифроваться независимо друг от друга или "со сцеплением" - когда результат шифрования текущего блока данных зависит от значения предыдущего блока или от результата шифрования предыдущего блока.
2. ***Поточное шифрование*** - необходимо, прежде всего, в тех случаях, когда информацию невозможно разбить на блоки - скажем, некий поток данных, каждый символ которых должен быть зашифрован и отправлен куда-либо, не дожидаясь остальных данных, достаточных для формирования блока. Поэтому алгоритмы поточного шифрования шифруют данные побитно или посимвольно.

Алгоритмы на основе сети Фейстеля.

.



Среди алгоритмов, основанных на сети ***Фейстеля***, можно привести в пример:

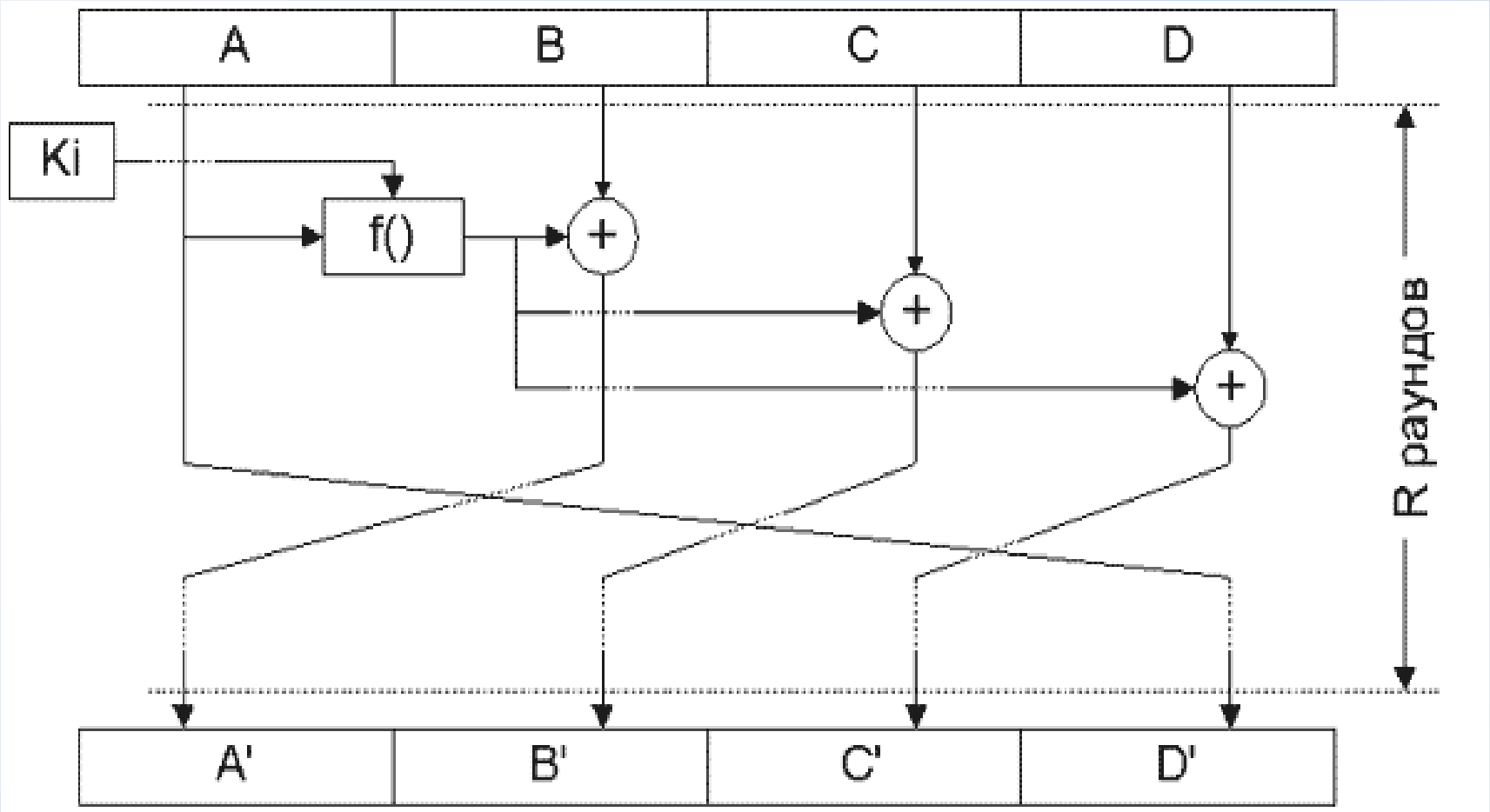
алгоритм DES (Data Encryption Standard)

ГОСТ 28147-89, RC5, Blowfish, TEA, CAST-128 и т.д.

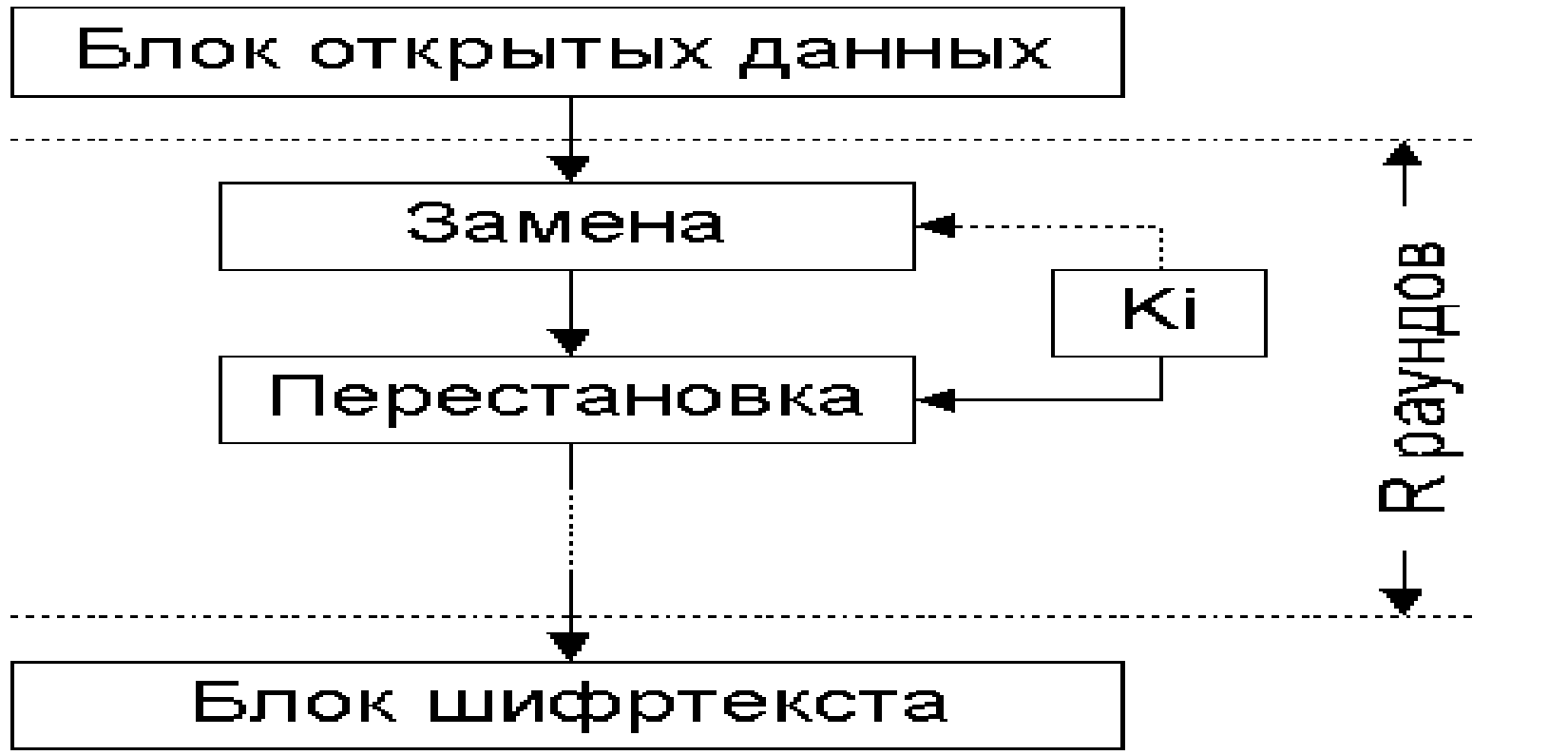
Панасенко С.П. Алгоритмы шифрования. Специальный справочник. – СПб.: БХВ-Петербург, 2009 – 576 с

На сети **Фейстеля** основано большинство современных алгоритмов шифрования

Обобщенная или *расширенная* сеть Фейстеля - алгоритм RC6.



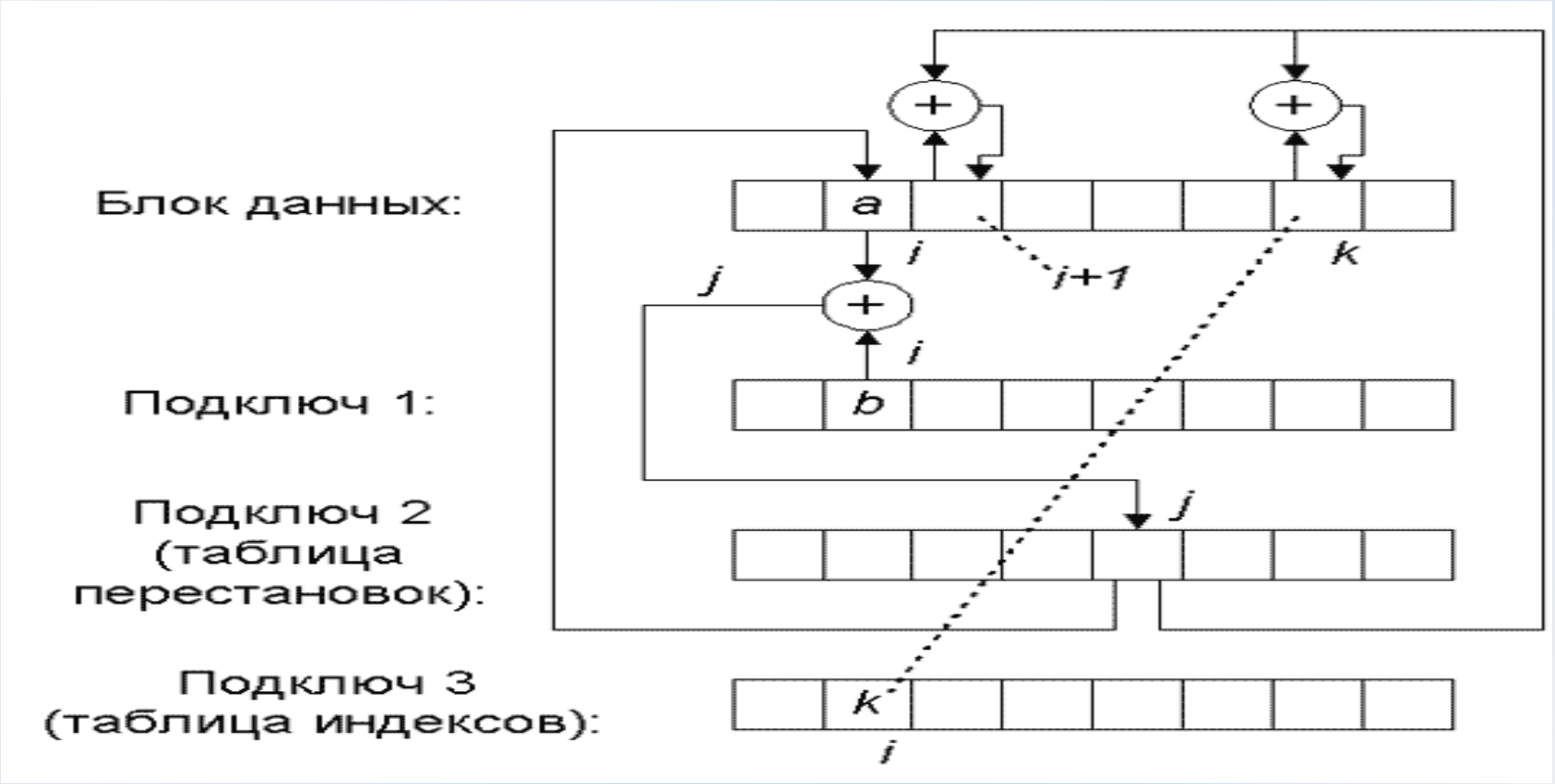
Подстановочно-перестановочные сети (*SP-сеть* - Substitution-permutation network)



алгоритмы Serpent или SAFER+

✓Алгоритмы со структурой *"квадрат"* (Square) - алгоритм Rijndael.

✓Алгоритмы с нестандартной структурой - алгоритм FROG



Длины ключей, достаточные для конфиденциальности

http://www.ecrypt.eu.org

Срок действия	Шифр с секретным ключом	RSA	Криптография на эллиптических кривых
дни/часы	50	512	100
5 лет	73	1024	146
10–20 лет	103	2048	206
30–50 лет	141	4096	282

Предположения: нет квантовых компьютеров; нет прорывов в науке; ограниченный бюджет.

Весьма редко встречаются алгоритмы шифрования, которые используют ключ шифрования (или его фрагменты) в «чистом» виде (таким алгоритмом является, например, отечественный стандарт шифрования ГОСТ 28147-89).

Подавляющее большинство алгоритмов шифрования выполняет существенную модификацию исходного ключа шифрования для его последующего использования в процессе преобразований.

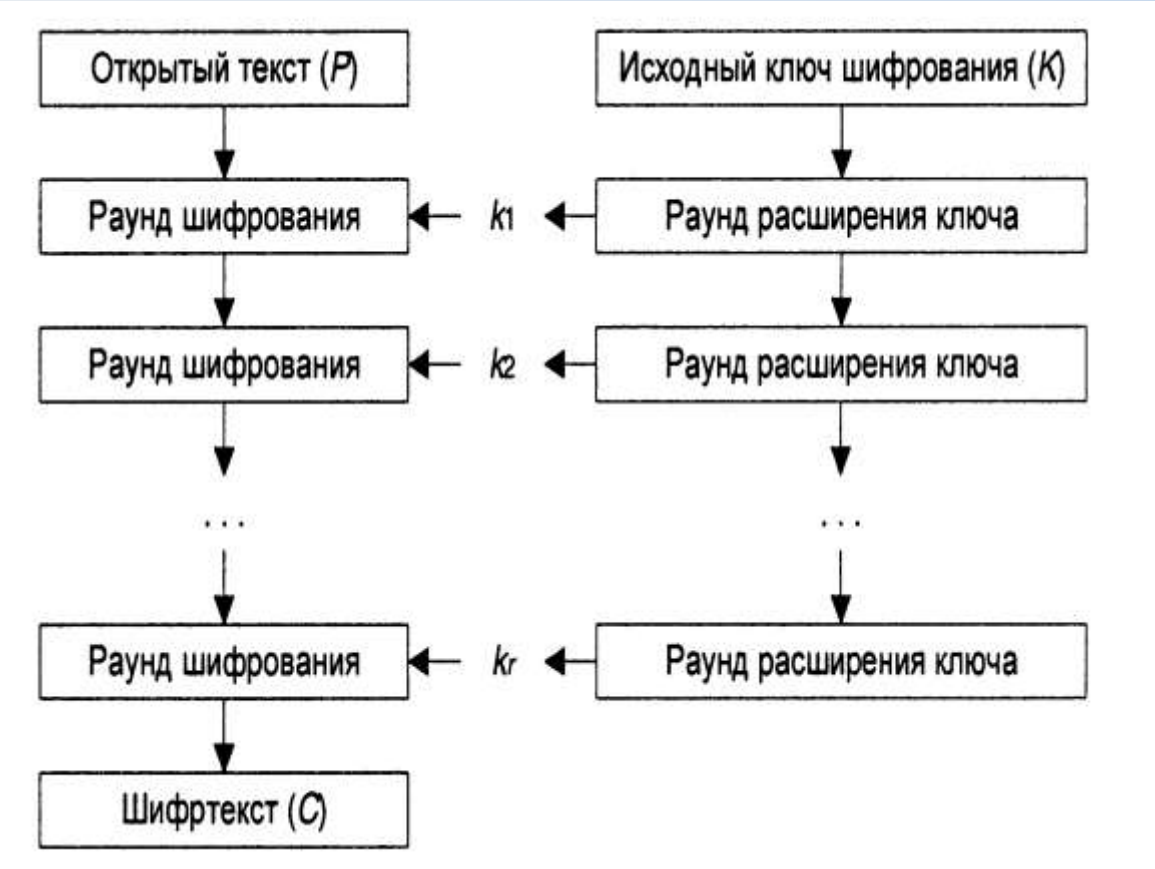
Такая модификация называется расширением ключа (key extension, key schedule);

Существуют примеры алгоритмов, в которых процедура расширения ключа является исключительно сложной по сравнению с собственно шифрованием, среди них стоит упомянуть алгоритмы HPC и FROG.

Название процедуры проистекает из того факта, что исходный ключ шифрования обычно имеет размер существенно меньший совокупности подключей, используемых в раундах алгоритма, т.е. *расширенного ключа*.

Получается, что алгоритм шифрования можно логически разделить на два субалгоритма:

- собственно шифрующие преобразования
- процедура расширения ключа



К процедуре расширения ключа предъявляется немало требований, целью которых является повышение криптостойкости и других характеристик алгоритма, например:

· Весьма желательно, чтобы процедура расширения ключа могла вычислять ключи «на лету» **(on-the-fly**), т.е. параллельно с шифрующими преобразованиями: это позволит как распараллеливать вычисления в многопроцессорных системах, так и не тратить память для хранения всего расширенного ключа при шифровании в условиях ограниченных ресурсов.

· В многих применениях алгоритмов симметричного шифрования (скажем, сетевой шифратор, использующий различные ключи для шифрования данных по различным направлениям или при использовании алгоритмов шифрования для построения хэш-функций) **часто приходится менять ключи** в шифраторе. Соответственно, весьма сложная процедура расширения ключа не позволит использовать алгоритм шифрования в данных случаях.

· Степень подверженности алгоритма атакам на связанных ключах также весьма зависит от процедуры расширения ключа.

Если удастся построить большой квантовый компьютер…

- Все схемы, основанные на задаче факторизации (такие, как RSA), станут нестойкими.
- То же самое для задачи дискретного логарифмирования (в том числе на эллиптических кривых).
- Длины ключей криптосистем с секретным ключом потребуется увеличить в два раза.
- Длины хэш-значений потребуется увеличить в полтора раза.

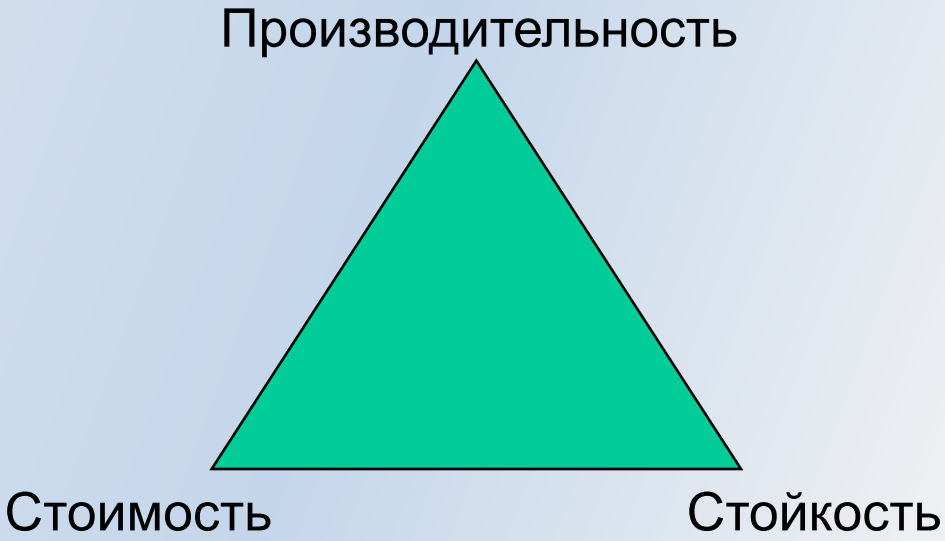
- Альтернативы: криптосистемы HFE, NTRU, …
- Пока представляется очень трудным обеспечить требуемую производительность современных систем, сохранив на должном уровне стойкость против обычных методов криптоанализа.

Основные нерешенные проблемы криптографии

- Обеспечение стойкости на период 50–100 лет.
- Аутентифицированное шифрование в сетях со скоростью передачи данных порядка Терабит/с.
- Создание криптографических схем с очень небольшим потреблением энергии и очень низкой ресурсоемкостью (ultra-low power/footprint).

Безопасная программная и аппаратная реализация криптографических схем на практике

Разработка легко переносимых алгоритмов (algorithm agility)



1. Barr, T **Invitation to Cryptology** Upper Saddle River.NJ: Prentice Hall, 2002
2. Bishop, D **Cryptography with Java Applets** Sudbury, MA: Jones and Bartlett, 2003
3. Bishop, M **Computer Security** Reading, MA: Addison-Wesley, 2005
4. Blahut, U **Algebraic Codes for Data Transmission** Cambridge:Cambridge University Press, 2003
5. Brassoud, D., and Wagon, S **Computational Number Theory.Emerville** CA: Key College, 2000
6. Coutinho, S **The Mathematics of Ciphers** Natick, MA: A. K.Peters, 1999
7. Dummit, D., and Foote, R **Abstract Algebra** Hoboken, NJ: John Wiley & Sons, 2004
8. Doraswamy, H., and Harkins, D **PSec. Upper Saddle River** NJ:Prentice Hall, 2003
9. Durbin, J **Modern Algebra** Hoboken, NJ: John Wiley & Sons,2005
10. Enge, A **Elliptic Curves and Their Applications to Cryptography** Norwell, MA: Kluver Academic, 1999
11. Forouzan, B **TCP/IP Protocol Suite** New York: McGraw-Hill,2006
12. Forouzan, B **Data Communication and Networking** New York:McGraw-Hill, 2007
13. Frankkel, S **Demystifying the IPSec Puzzle** Norwood, MA:Artech House, 2001
14. Garret, P **Making, Breaking Codes** Upper Saddle River, NJ:Prentice Hall, 2001
15. Kahn, D **The Code breakers: The Story of Secret Writing** New York: Scribner,1996
16. Kaufman, C., Perlman, R., and Speciner, M **Network Security** Upper Saddle River, NJ: Prentice Hall, 2001
17. Larson, R., Edwards, B., and Falvo, D **Elementary Linear Algebra** Boston: Houghton Mifflin, 2004
18. Mao, W **Modern Cryptography** Upper Saddle River, NJ: Prentice Hall, 2004
19. Menezes, A., Oorschot, P., and Vanstone, S **Handbook of Applied Cryptography** New York: CRC Press, 1997
20. Pieprzyk, J., Hardjono, T., and Seberry, J **Fundamentals of Computer Security** Berlin: Springer, 2003
21. Rescoria, E **SSL and TLS** Reading, MA: Addison-Wesley, 2001
22. Rhee, M **Internet Security** Hoboken, NJ: John Wiley & Sons,2003
23. Rosen, K **Elementary Number Theory** Reading, MA: Addison-Wesley, 2006
24. Solomon, D **Data Privacy and Security** Berlin: Springer, 2003
25. Schneier, B **Applied Cryptography** Reading, MA: Addison-Wesley,1996
26. Stallings, W **Cryptography and Network Security** Upper Saddle River, NJ: Prentice Hall, 2006
27. Stinson, D **Cryptography: Theory and Practice** New York: Chapman & Hall/CRC, 2006
28. Thomas, S **SSL and TLS Essentials** New York: John Wiley & Sons, 2000
29. Trappe, W., and Washington, L **Introduction to Cryptography and Coding Theory** Upper Saddle River, NJ: Prentice Hall, 2006
30. Vaudenay, S **A Classical Introduction to Cryptography** New York: Springer, 2006

На сайте ***http://www.intuit.ru***

•Анисимов А.А.
Менеджмент в сфере информационной безопасности
БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий - ИНТУИТ.ру, 2009

•Галатенко В.А.
Основы информационной безопасности
Интернет-университет информационных технологий - ИНТУИТ.ру, 2008

•Лапонина О.Р.
Основы сетевой безопасности: криптографические алгоритмы и протоколы взаимодействия
Интернет-университет информационных технологий - ИНТУИТ.ру, 2005

•Галатенко В.А.
Стандарты информационной безопасности
Интернет-университет информационных технологий - ИНТУИТ.ру, 2005