**Вопросы к экзамену по дисциплине «Основы защиты информации»**

**4-й курс направления «программная инженерия»**

**январь 2022**

1. Информация и ее свойства, связанные с обеспечением безопасности. Качество информации. Утечка информации.
2. Конфиденциальность, целостность, доступность информации.
3. Угрозы информационной безопасности. Источники угроз.
4. Цели и средства защиты информации.
5. Классификация методов защиты информации.
6. Стандарты информационной безопасности.
7. Уязвимости ПО. Безопасное программирование.
8. Программная закладка.
9. Скрытый канал.
10. Недокументированные возможности.
11. Сканеры уязвимостей.
12. Обнаружение и предотвращение вторжений.
13. Контроль доступа субъекта.
14. Идентификация. Аутентификация. Авторизация.
15. Разграничение доступа.
16. Криптография. Криптоанализ. Криптология.
17. Симметричное шифрование. Шифр Вижинера. Шифр Цезаря. Пропорциональный шифр. Шифр перестановки.
18. “Вскрытие” симметричных шифров. Усиление симметричных шифров.
19. Сеть Фейстеля. Блочное шифрование.
20. Требования к алгоритму шифрования.
21. DES.
22. ГОСТ-28147-89.
23. Понятие хеш-функции.
24. Требования к криптохеш-функциям.
25. Использование блочных алгоритмов шифрования для формирования хеш-функции.
26. MD5.
27. Семейство алгоритмов SHA.
28. ГОСТ 34.11-2018.
29. Принцип поточного шифрования. Гаммирование.
30. ГСЧ и ГПСЧ.
31. Криптосвойства ГПСЧ.
32. Линейный конгруэнтный ГПСЧ.
33. Метод Фибоначчи с запаздываниями (Lagged Fibonacci Generator).
34. ГПСЧ на основе алгоритма BBS.
35. ГСПЧ на основе сдвиговых регистров с обратной связью.
36. ГСЧ.
37. Симметричное шифрование. Односторонние функции.
38. Алгоритм шифрования Эль-Гамаля.
39. Криптосистема Меркла-Хеллмана.
40. RSA.
41. Алгоритм обмена ключа Диффи-Хеллмана.
42. Информация и ее свойства, связанные с обеспечением безопасности. Качество информации. Утечка информации.

**Информация** - это любое представление данных, которое имеет ценность и может быть использовано для принятия решений, планирования или осуществления действий. Это может быть что угодно, от секретных правительственных планов до личных данных пользователя.

**Свойства информации**, связанные с обеспечением безопасности:

* **Конфиденциальность**: Означает, что информация доступна только авторизованным лицам или системам. Нарушение конфиденциальности приводит к несанкционированному доступу к информации.
* **Целостность**: Означает, что информация является полной, точной и не подвергалась несанкционированной модификации. Нарушение целостности означает, что информация повреждена, искажена или изменена без разрешения.
* **Доступность**: Означает, что авторизованные лица или системы могут получить доступ к информации, когда это необходимо. Нарушение доступности приводит к невозможности использования информации.
* **Подлинность** (Аутентичность): Означает, что информация получена из достоверного источника и не является подделкой. Подтверждение подлинности критически важно для принятия обоснованных решений.
* **Неотказуемость** (Неотрекаемость): Означает, что отправитель информации не может отрицать факт ее отправки, а получатель не может отрицать факт ее получения. Это важно для обеспечения юридической силы транзакций и других операций.

**Качество информации** влияет на ее ценность и на то, насколько эффективно она может быть использована. Вот некоторые ключевые атрибуты качества информации:

* **Точность**: Насколько информация соответствует действительности.
* **Полнота**: Содержит ли информация все необходимые данные.
* **Актуальность**: Насколько информация своевременна и соответствует текущему моменту.
* **Релевантность**: Насколько информация соответствует потребностям пользователя.
* **Понятность**: Насколько легко информация воспринимается и интерпретируется.
* **Согласованность**: Насколько информация согласована с другими источниками информации.

**Утечка информации:**

Утечка информации - это несанкционированное раскрытие конфиденциальной информации лицам или системам, не имеющим права доступа к ней. Утечка может происходить разными путями:

* **Технические каналы**: Взлом компьютерных систем, перехват сетевого трафика, использование вредоносного программного обеспечения (вирусов, троянов), несанкционированный доступ к базам данных.
* **Человеческий фактор:** Небрежность сотрудников (например, отправка конфиденциальной информации по электронной почте без шифрования), социальная инженерия (манипулирование людьми с целью получения информации), подкуп, шпионаж.
* **Физические каналы**: Кража или потеря носителей информации (ноутбуков, USB-накопителей, документов), несанкционированный доступ к физическим объектам (офисам, серверам), прослушивание помещений.
* **Анализ побочных данных:** Получение информации из метаданных, шаблонов использования, анализа сетевого трафика, и других непрямых источников.

1. Конфиденциальность, целостность, доступность информации.

Конфиденциальность, целостность и доступность (ЦЦД) - основа информационной безопасности.

* **Конфиденциальность**: Защита от несанкционированного доступа к информации (кто не должен знать, не должен получить доступ). Обеспечивается шифрованием, контролем доступа, политиками безопасности.
* **Целостность**: Обеспечение точности и полноты информации, защита от несанкционированной модификации (никто не должен изменить информацию без разрешения). Обеспечивается контролем версий, резервным копированием, хешированием.
* **Доступность**: Обеспечение своевременного и надежного доступа к информации авторизованным пользователям (информация должна быть доступна тогда, когда она нужна). Обеспечивается отказоустойчивостью, балансировкой нагрузки, планами аварийного восстановления.

1. Угрозы информационной безопасности. Источники угроз.

**Угроза информационной безопасности:** Потенциальное событие, действие или бездействие, которое может нанести ущерб информации, системам и ресурсам.

**Основные типы угроз:**

* Конфиденциальность: Несанкционированный доступ к информации.
* Целостность: Искажение или уничтожение информации.
* Доступность: Блокирование доступа к информации или ресурсам.

**Основные источники угроз:**

* Злоумышленники: Хакеры, конкуренты, криминальные группы, инсайдеры.
* Пользователи: Ошибки, халатность, некомпетентность.
* Техногенные факторы: Ошибки ПО, аппаратные сбои, перебои в электроснабжении.
* Стихийные факторы: Природные катаклизмы.

**Ключевые примеры угроз:**

* Вирусы и вредоносное ПО.
* Фишинг и социальная инженерия.
* DDoS-атаки.
* Взлом учетных записей.
* Кража оборудования.
* Утечки информации.

1. Цели и средства защиты информации.

Цели:

* Снижение вероятности техногенных угроз,
* Снижение ущерба от стихийных явлений,
* Снижение вероятности угроз, реализуемых по причине халатности или недостаточной квалификации,
* Защита от нарушителей, действующих из любопытства или самоутверждения,
* Защита от нарушителей, действующих злонамеренно.

Средства:

* Аппаратные (межсетевые экраны, IDS/IPS, шифрование)
* Программные (антивирусы, контроль доступа, резервное копирование)
* Организационные (политики, обучение, управление доступом)
* Физические (ограничение доступа, видеонаблюдение)

1. Классификация методов защиты информации.



1. Стандарты информационной безопасности.

* ГОСТ 28147-89 – стандарт шифрования государственной тайны,
* ГОСТ 34.11—2018 – стандарт хэширования,
* ГОСТ 34.10-2018 – стандарт цифровой подписи,
* ГОСТ Р 50739-95 – стандарт защиты от несанкционированного доступа к информации.
* ГОСТ 58142 «Информационная технология. Методы и средства обеспечения безопасности. Детализация анализа уязвимостей программного обеспечения»

УК РФ Глава 28. Преступления в сфере компьютерной информации

1. Уязвимости ПО. Безопасное программирование.

* Уязвимость ПО - недостаток программно-технического средства или информационной системы в целом, который может быть использован для реализации угроз безопасности информации.
  + Уязвимость кода,
  + Уязвимость конфигурации,
  + Уязвимость архитектуры,
  + Организационная уязвимость,
  + Многофакторная уязвимость.

**ГОСТ** Р 56939-2016 Разработка **безопасного** программного обеспечения.

Методика разработки программного обеспечения, предотвращающая случайное внедрение уязвимостей и обеспечивающая устойчивость к воздействию вредоносного ПО и несанкционированному доступу.

* Defensive programming(оборонительное программирование)
* Secure coding(безопасное программирование)

1. Программная закладка.

**Закла́дка** (**программная закладка**) — скрытно внедрённая в защищенную систему программа, либо намеренно измененный фрагмент программы, который позволяет злоумышленнику осуществить несанкционированный доступ к ресурсам системы на основе изменения свойств системы защиты. Закладка может быть внедрена самим разработчиком программного обеспечения.

Часто программные закладки выполняют роль перехватчиков паролей, трафика, а также служат в качестве проводников для компьютерных вирусов.

Программные закладки невозможно обнаружить при помощи стандартных антивирусных средств, их выявление возможно только специальными тестовыми программами.

1. Скрытый канал.

Скрытый канал – это непредусмотренный разработчиком системы информационных технологий и автоматизированных систем коммуникационный канал, который может быть применен для нарушения политики безопасности.

С помощью скрытых каналов могут быть реализованы следующие нарушения политики безопасности:

* Угроза внедрения вредоносных программ и данных .
* Угроза подачи нарушителем команд агентом для выполнения его функций .
* Угроза утечки криптографических ключей, паролей (несанкционированный доступ к ним) или отдельных информационных объектов .

Выявление

* ***статистический метод*** - подразумевает сбор статистических данных о пакетах, проходящих через защищаемый участок сети, без внесения в них каких-либо изменений.
* ***сигнатурный метод*** - аналогичен способу, используемому антивирусным ПО для поиска вредоносных программ. При наличии набора известных реализаций скрытых каналов, для каждой из них формируется сигнатура. В потоке данных проводится поиск таких сигнатур.

1. Недокументированные возможности.

Это возможности устройств или ПО, не описанные в официальной документации. Они могут быть:

• Сознательно заложены разработчиками (для тестирования, расширения, совместимости, скрытого контроля).

• Возникнуть случайно как побочные эффекты.

НДВ отличаются от скрытых, но задокументированных возможностей. Их обычно находят через обратную разработку или случайно.

1. Сканеры уязвимостей.

Программы, которые автоматически ищут известные уязвимости в программном обеспечении, сетевых устройствах и веб-приложениях.

Зачем нужны:

* Поиск слабых мест: Обнаружение уязвимостей до того, как ими воспользуются злоумышленники.
* Оценка рисков: Определение приоритетных областей для усиления безопасности.
* Соответствие стандартам: Соблюдение требований безопасности.
* Автоматизация тестирования: Ускорение процесса проверки безопасности.

Web:

* OWASP ZAP
* W9scan
* Wapiti
* Arachni
* Paros
* Tenable.io
* Burp Suite Pro
* Acunetix

1. Обнаружение и предотвращение вторжений.

Комплекс мер и инструментов, направленных на выявление и блокировку несанкционированного доступа или вредоносной активности в компьютерных системах и сетях.

Две основные категории:

* Системы обнаружения вторжений (IDS):
  + Что делают: Мониторят трафик и активность в системе/сети на предмет подозрительного поведения.
  + Как работают: Используют сигнатурный анализ (сопоставление с известными атаками) и аномальный анализ (выявление отклонений от нормального поведения).
  + Действия: Обнаруживают, регистрируют и оповещают об инцидентах, но не блокируют их активно. Это значит, что они информируют о проблеме, но не предпринимают действий для ее устранения.
* Системы предотвращения вторжений (IPS):
  + Что делают: Расширенная версия IDS, которая не только обнаруживает, но и блокирует или смягчает атаки в реальном времени.
  + Как работают: Аналогично IDS, но имеют возможность автоматически реагировать на обнаруженные угрозы.
  + Действия: Блокируют трафик, закрывают порты, сбрасывают соединения, карантинируют файлы, и т.д. Активно предотвращают дальнейшее развитие атаки.

UTM — это универсальный пакет утилит, сочетающий в себе множество мелких модулей защиты. UTM бывают программными или аппаратными и, как правило, включают в себя сразу IDS, IPS, файервол, а зачастую и антивирус, прокси-сервер, почтовые фильтры, VPN и т.д.

1. Контроль доступа субъекта.
2. Идентификация. Аутентификация. Авторизация.

**Идентификация** - это процесс установления соответствия между предъявленным субъектом и одним из зарегистрированных идентификаторов в информационной системе. Иными словами, это процедура, посредством которой субъект заявляет о своей принадлежности к определенной учетной записи или профилю, известному системе.

Под **аутентификацией** понимается процедура установления подлинности предъявляемого субъектам идентификатора, т.е. задача именно установления принадлежности идентификатора именно этому субъекту.

**Авторизация** - это предоставление субъекту полномочий, связанных с его идентификатором в информационной системе.

То есть после того, как информационная система удостоверилась в том, что пользователь является именно тем, за кого он себя выдает, система предоставляет ему те права, которые в этой системе данному пользователю предоставлены



1. Разграничение доступа.

**Разграничение доступа** - это механизм контроля, определяющий, кто имеет право получать, изменять или удалять информацию и ресурсы системы. Цель - предотвратить несанкционированный доступ и защитить конфиденциальность, целостность и доступность данных.

* Дискреционное (DAC): Владелец ресурса определяет, кто и что может с ним делать. Гибко, но потенциально небезопасно (риск утечки данных из-за ошибок владельцев).
* Мандатное (MAC): Система (а не владелец) контролирует доступ на основе меток безопасности (классификация). Очень строго, обеспечивает высокую безопасность, но менее гибко.
* Ролевое (RBAC): Доступ назначается ролям, а роли назначаются пользователям. Упрощает управление доступом в больших организациях, обеспечивает баланс между безопасностью и удобством.

1. Криптография. Криптоанализ. Криптология.

**Криптология**— наука, занимающаяся методами шифрования и расшифровывания.

**Криптография** занимается преобразованием исходного текста в кажущуюся случайной последовательность символов, называемую криптограммой или шифротекстом.

**Стойкость криптоалгоритма** определяется сложностью вычислений алгоритмов дешифрования.

**Криптоанализ** – наука, которая занимается разработкой методов дешифрования криптограмм, при неизвестном алгоритме дешифрования и ключе.

1. Симметричное шифрование. Шифр Вижинера. Шифр Цезаря. Пропорциональный шифр. Шифр перестановки.

**Симметричное шифрование** — способ шифрования, в котором для шифрования и дешифрования применяется один и тот же криптографический ключ.

**Пропорциональный шифр (кратко):**

Определение: Шифр подстановки, в котором каждая буква открытого текста заменяется на другую букву с использованием математической формулы (C = (a × P + b) mod m).

Как работает:

* Буква -> Число.
* Шифрование: (a × Число + b) mod размер\_алфавита = Зашифрованное число. C = (a \* P + b) mod m
* Дешифрование: мультипликативный\_обратный(a) × (Зашифрованное число - b) mod размер\_алфавита = Исходное число. P = a^-1 \* (C - b) mod m
* Число -> Буква.

Ключ состоит из a и b, где a должно быть взаимно простым с размером алфавита.

**Шифре Вижинера:**

Это полиалфавитный шифр (использует несколько шифров Цезаря) для повышения безопасности. Он использует ключевое слово, чтобы сдвигать каждую букву сообщения на разное количество позиций в алфавите.

Пример:

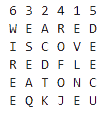
* Открытый текст: "HELLO"
* Ключевое слово: "KEY"

**Шифр перестановки:**

Шифр перестановки (или транспозиции) - это метод шифрования, при котором буквы открытого текста не заменяются, а просто меняются местами по определенному правилу. Ключ определяет, как именно переставляются буквы.

В процессе шифрования сообщение записывается в виде таблицы. Количество колонок таблицы определяется размером ключа. Например, зашифруем сообщение WE ARE DISCOVERED. FLEE AT ONCE с помощью ключа 632415.

Так как ключ содержит 6 цифр дополним сообщение до длины кратной 6 произвольно выбранными буквами QKJEU и запишем сообщение в таблицу, содержащую 6 колонок, слева направо:



1. “Вскрытие” симметричных шифров. Усиление симметричных шифров.

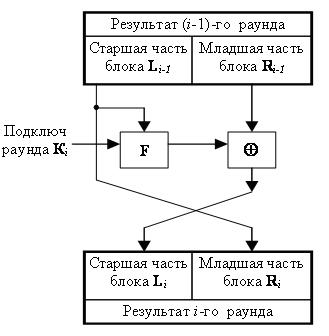
Применить композиционный шифр: сначала шифр Цезаря, затем перестановки

На стойкость шифра влияют такие факторы, как размер блока, размер ключа, количество раундов шифрования.

Алгоритмы симметричного шифрования могут обрабатывать исходный текст блоками или потоком.

1. Сеть Фейстеля. Блочное шифрование.

**Блочное шифрование** - это метод шифрования, при котором данные разделяются на блоки фиксированной длины, и каждый блок шифруется отдельно с использованием одного и того же ключа. В отличие от потокового шифрования, где шифруется каждый бит или байт данных по отдельности, блочное шифрование оперирует с большими блоками данных, что позволяет реализовывать более сложные преобразования и достигать большей безопасности.



1. Требования к алгоритму шифрования.

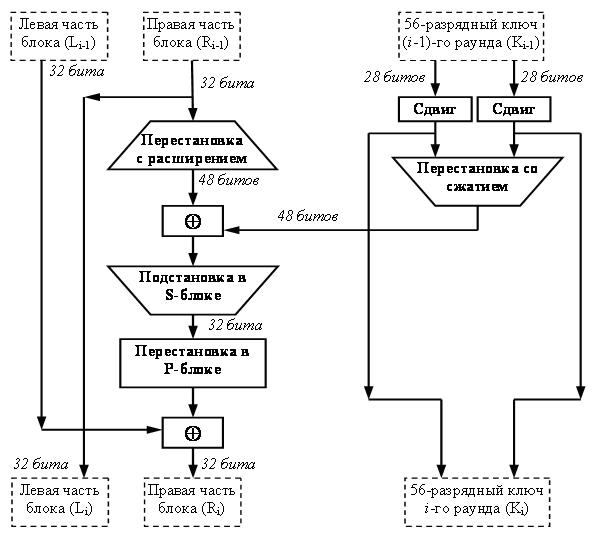
Алгоритм шифрования должен быть:

* Безопасным: Стойким к атакам, иметь лавинный эффект, ключевую зависимость.
* Быстрым: Эффективно шифровать и расшифровывать данные.
* Практичным: Легко реализуемым, стандартизированным, масштабируемым.

1. DES.

DES – Data Encryption Standard

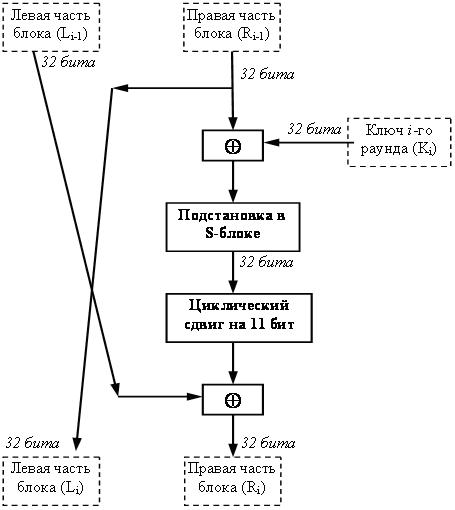
Размер блока 64 бита, длина ключа 56 бит, количество раундов – 16.

1. ГОСТ-28147-89.

длинный *ключ* – 256 *бит,* 32 раунда шифрования, размер блока составляет 64 бита,

*Алгоритм* представляет собой классическую *сеть* Фейстеля.



1. Понятие хеш-функции.
2. Требования к криптохеш-функциям.

Требования:

* Можно применить к сообщению любого размера
* Быстрое вычисление хеш
* Трудно восстановить/подобрать исходное сообщение по известному хеш
* Подобрать сообщения с одинаковым хеш – трудно/долго
* Вероятность возникновения коллизий минимальна

1. Использование блочных алгоритмов шифрования для формирования хеш-функции.
2. MD5.

**Шаг 1. Выравнивание потока** – дописывание в конец потока 1 и необходимого количества нулей, чтобы деление длины сообщения по модулю 512 дало результат 448.

Добавление производится всегда, даже если сообщение имеет нужную длину.

**Шаг 2. Добавление длины сообщения –** дописывается 64-битное представление длины исходного сообщения.

Длина сообщения кратна 512 битам – кратна шестнадцати 32-битным словам.

**Шаг 3. Инициализация буфера, функций и белого шума**

Шаг 4. Перестановки(4 раунда)

1. Семейство алгоритмов SHA.

Семейство алгоритмов SHA (Secure Hash Algorithm) - это группа криптографических хеш-функций, разработанных Национальным институтом стандартов и технологий США (NIST). Они используются для создания "отпечатка пальца" или хеша данных, который представляет собой строку фиксированной длины, вычисленную на основе входных данных.

Семейство SHA эволюционировало с течением времени, чтобы противостоять растущим вычислительным мощностям и атакам.

* SHA-0: Самая первая версия, опубликованная в 1993 году. Вскоре после публикации были обнаружены недостатки, и алгоритм был быстро заменен на SHA-1. Не рекомендуется к использованию.
* SHA-1: Опубликован в 1995 году. Генерирует хеш длиной 160 бит. Долгое время был широко распространен, но со временем появились успешные атаки, демонстрирующие возможность нахождения коллизий. К настоящему времени считается небезопасным и не рекомендуется к использованию в новых системах.
* SHA-2: Опубликован в 2001 году. Включает в себя несколько вариантов, различающихся длиной генерируемого хеша:
* SHA-224: 224-битный хеш
* SHA-256: 256-битный хеш
* SHA-384: 384-битный хеш
* SHA-512: 512-битный хеш
* SHA-3: Опубликован в 2015 году. Не является модификацией SHA-2, а представляет собой совершенно новый алгоритм, основанный на конструкции "губки" (sponge construction) и названный Keccak. Хотя SHA-2 все еще считается достаточно безопасным, SHA-3 был разработан как альтернатива на случай, если в SHA-2 будут обнаружены серьезные уязвимости. Как и SHA-2, SHA-3 включает в себя несколько вариантов с разными длинами хеша:
* SHA3-224: 224-битный хеш
* SHA3-256: 256-битный хеш
* SHA3-384: 384-битный хеш
* SHA3-512: 512-битный хеш

1. ГОСТ 34.11-2018.Стрибог

Имеет две реализации с результирующим значением длиной 256 или 512 бит.

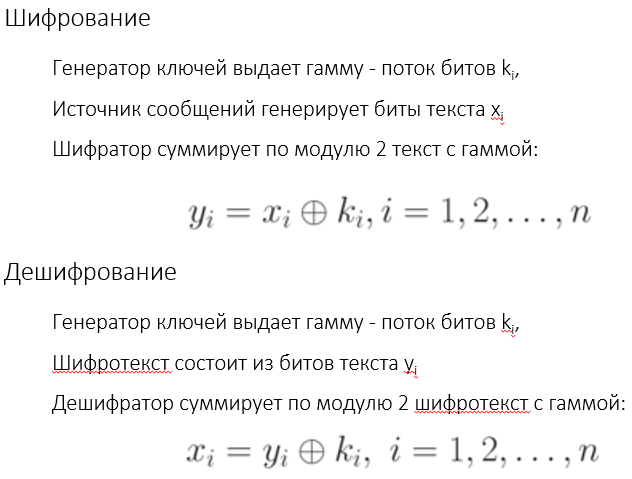
Если длина сообщения больше 512 бит (или 64 байт), то оно делится на блоки по 512 бит, а оставшийся кусочек дополняется нулями с одной единичкой до 512 бит (или до 64 байт). Если длина сообщения меньше 512 бит, то оно сразу дополняется нулями с единичкой до полных 512 бит.

Основу хеш-функции «Стрибог» составляет функция сжатия, построенная на блочном шифре.

1. Принцип поточного шифрования. Гаммирование.

**Гамми́рование** — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст.

z = (x + k) mod N



1. ГСЧ и ГПСЧ.

Для генерации ключей используются генераторы псевдослучайных чисел(ГПСЧ) и генераторы случайных чисел(ГСЧ).

Случайные числа – возможно генерировать измеряя случайные величины, которые называются шумом. В результате измерения шума в выборке и перевода его в числа, получается случайная числовая последовательность, которая никогда не повторится.

Псевдослучайные числа – генерируются на основе алгоритмов, всегда можно указать период, через который числа начнут повторяться.

* ГПСЧ используются в качестве генераторов ключей в поточных шифрах.
* Целью использования *ГПСЧ* является получение "бесконечного" ключевого слова, располагая относительно малой длиной самого ключа.
* *ГПСЧ* создает последовательность битов, похожую на случайную.

Такие последовательности вычисляются по определенным правилам и не являются случайными => они могут быть абсолютно точно воспроизведены как на передающей, так и на принимающей стороне.

1. Криптосвойства ГПСЧ.

период последовательности должен быть очень большой;

порождаемая последовательность должна быть "почти" неотличима от действительно случайной;

вероятности порождения различных значений должны быть в точности равны;

вычисление числа ki+1по известным предыдущим элементам последовательности ki без знания секретного ключа должно быть трудной задачей.

1. Линейный конгруэнтный ГПСЧ.

* ki=(a\*ki-1+b)mod c,

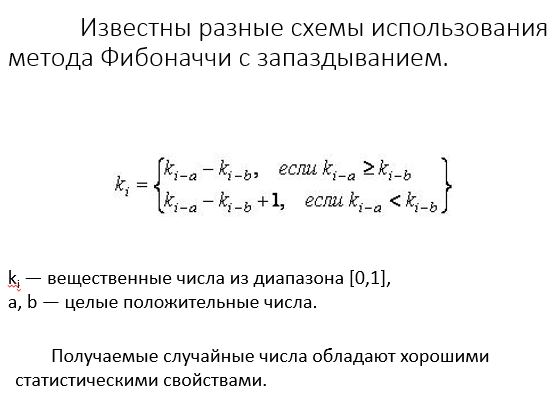
а, b, с — некоторые константы,

ki-1 — предыдущее псевдослучайное число.

Для получения k1 задается начальное значение k0.

* Достоинством линейных конгруэнтных *генераторов псевдослучайных чисел* является их простота и высокая скорость получения псевдослучайных значений.

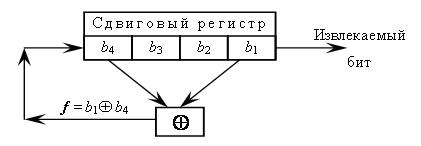
1. Метод Фибоначчи с запаздываниями (Lagged Fibonacci Generator).



1. ГПСЧ на основе алгоритма BBS.

* *генератор с квадратичным остатком.*
* Вначале выбираются два больших простых числа p и q. Числа p и q должны быть оба *сравнимы* с 3 по модулю 4, то есть при делении p и q на 4 должен получаться одинаковый *остаток* 3.
* Далее вычисляется число M = p\* q, называемое целым числом Блюма.
* Затем выбирается другое случайное *целое число* х, взаимно простое с М.
* Вычисляем х0= х2mod M. х0 называется стартовым числом генератора.
* На каждом n-м шаге работы генератора вычисляется
* хn+1= хn2 mod M.
* Результатом n-го шага является один (обычно младший) *бит* числа хn+1. Иногда в качестве результата принимают *бит* чётности, то есть количество единиц в двоичном представлении элемента. Если количество единиц в записи числа четное – *бит* четности принимается равным 0, нечетное – *бит* четности принимается равным 1.

1. ГСПЧ на основе сдвиговых регистров с обратной связью.



1. ГСЧ.

* компонент Firmware Hub для чипсетов - регистрирует тепловой шум с резисторов, усиливает его и использует результирующий сигнал для изменения периода относительно медленного генератора тактовых импульсов.
* регистрация точного времени нажатия на клавиши использовалась как удобный источник случайных стартовых значений для генераторов в прошлом. Для тех же целей использовали передвижения мыши и даже скорость поиска секторов на жёстком диске.
* шум звуковой карт, такты процесса, размер свободной памяти жесткого диска…

1. Симметричное шифрование. Односторонние функции.
2. Алгоритм шифрования Эль-Гамаля.

Асимметричный криптографический алгоритм, основанный на трудности решения задачи дискретного логарифмирования в конечном поле.

1. Криптосистема Меркла-Хеллмана.
2. RSA.
3. Выбираются два различных случайных простых числа p и q.
4. Вычисляется их произведение n=p⋅q, которое называется *модулем*.
5. Вычисляется значение функции Эйлера от числа n:

φ(n)=(p−1)⋅(q−1).

1. Выбирается целое число e (1<e<φ(n), взаимно простое со значением функции  φ(n)φ(n). Обычно в качестве e берут простые числа, содержащие небольшое количество единичных бит в двоичной записи. Число e называется открытой экспонентой.
2. Вычисляется число d, удовлетворяющее сравнению:

d⋅e≡1(modφ(n)).

Число d называется секретной экспонентой.

1. Пара {e,n} публикуется в качестве открытого ключа RSA.
2. Пара {d,n} играет роль закрытого ключа RSA и держится в секрете.
3. Для шифрования сообщения m необходимо выполнить

m=(cd)mod n

1. Для шифрования сообщения m необходимо выполнить

c=(me)mod n

1. Пример шифрования RSA–
2. Алгоритм обмена ключа Диффи-Хеллмана.