## ГИБРИДНЫЕ КРИПТОСИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ

Сильные и слабые стороны симметричных и несимметричных криптографических систем защиты информации.

**Достоинства** симметричных криптосистем.

1. Высокая скорость реализации криптографической защиты информации Аппаратные средства способны шифровать данные со скоростью десятков Мбайт/с, программные средства - единиц Мбайт/с.
2. Известны симметричные системы шифрования, являющиеся безусловно стойкими криптосистемами.
3. Ключи для симметричных криптосистем сравнительно невелики (десятки - сотни бит), что упрощает их формиро-вание хранение, доставку, использование и замену.
4. Симметричные системы могут быть использованы для построения составной криптосистемы для увеличения ее криптографической стойкости.
5. Симметричные системы шифрования удобны для информационного обмена в сетях, особенно в режиме

6. Симметричные криптосистемы исследовались в течение десятков и сотен лет, и их безопасность многократно проверялась за время их практического использования.

**Недостатки симметричных криптосистем**:

1. Необходимость обеспечения секретности ключей у всех корреспондентов (пользователей) симметричных криптосистем.
2. Для больших информационных систем возникают трудноразрешимые проблемы безопасной доставки ключей участникам информационного обмена. 3. В информационных сетях со многими корреспондентами необходима частая смена одинаковых для всех ключей (так как криптосвязность обеспечивается, как правило, по принципу общей ключевой сети).
4. Симметричные криптосистемы не способны обеспечить аутентификацию сообщений и объектов в условиях взаимного недоверия и возможного обмана со стороны участников информационного обмена.

**Достоинства несимметричных криптосистем**:

1. Нет необходимости обеспечения секретности всей ключевой информации в несимметричных криптосистемах.
2. Проще решаются вопросы доставки ключей участникам информационного обмена.
3. Не требуется частой смены ключей пользователей несимметричных криптосистем (так как криптосвязность обеспечивается, как правило, по принципу ключевого направления или сети циркулярной связи).
4. Несимметричные криптосистемы способны обеспечить аутентификацию сообщений и объектов в условиях взаимного недоверия и возможного обмана со стороны участников информационного обмена, причем ключи проверки могут быть очень короткими.
5. В больших информационных сетях для обеспечения криптосвязности участников информационного обмена количество ключей для несимметричных криптосистем может быть значительно меньше, чем для симметричных.

**Недостатки несимметричных криптосистем**:

1. Достижимые скорости реализации криптографической защиты информации в несимметричных криптосистемах на несколько порядков ниже, чем в симметричных криптосистемах.
2. Размеры ключей для несимметричных криптосистем, как правило, существенно больше, чем для симметричных криптосистем.
3. Современные несимметричные криптосистемы не являются безусловно стойкими, более того, неизвестно ни одной практически реализуемой несимметричной системы шифрования, стойкость которой была бы строго доказана при любых атаках нарушителя.
4. Безопасность известных практически используемых несимметричных криптосистем основана на вычислительной сложности узкого класса теоретико-числовых задач, для которых в будущем могут быть найдены эффективные алгоритмы их решения (взлома криптосистем).

5. Безопасность несимметричных криптосистем исследовалась в течение сравнительно малого времени (двадцать лет с момента их возникновения), что увеличивает вероятность выявления их неизвестных пока недостатков.

Симметричные и несимметричные криптосистемы защиты информации имеют взаимодополняющие достоинства. Симметричные криптосистемы наиболее эффективны для сохранения в тайне защищаемой информации, а несиммет-ричные - для контроля ее подлинности и установления криптосвязности корреспондентов.

Поэтому на практике используют комбинированные криптографические системы защиты информации, называющиеся гибридными криптосистемами.

Примером гибридной криптосистемы защиты информации является популярная в сети Интернет программно-реализованная криптосистема зашиты информации PGP (Pretty Good Privacy), разработанная коллективом специалистов во главе с Ф. Циммерманом.

Программа PGP обеспечивает шифрование передаваемых по сети данных, подтверждение подлинности сообщений с помощью цифровой подписи, формирование и обмен ключевой информации пользователей.

Для шифрования данных используется алгоритм блочного шифрования IDEA (International data encryption algorithm), разработанный в конце 80-х годов известным криптографом Дж Месси. Как и DES, IDEA шифрует блоки данных длиной 64 бита, но использует более длинные 128-битные ключи, что практически исключает при атаке нарушителя перебор всех ключей алгоритма.

Перед шифрованием данных в PGP рекомендуется их сжимать с помощью алгоритма сжатия типа ZIP.

Сжатие избыточной информации целесообразно как с точки зрения снижения требований к пропускной способности используемых каналов связи, так и для существенного повышения криптостойкости алгоритмов шифрования благодаря удалению избыточности шифруемой информации

Целостность и авторство передаваемой в PGP информации обеспечивается использованием цифровой подписи сообщений по алгоритму RSA. Пользователь PGP может сам устанавливать требуемую степень криптографической стойкости алгоритма цифровой подписи, выбирая длину ключа подписи or нескольких сотен до 1024 бит.

В качестве криптографической функции хеширования при подписи сообщения используется функция MD5, сжимающая подписываемое сообщение произвольного размера в хэш-код длиною в 128 бит. В PGP нетрадиционно построена система управления криптографическими ключами пользователей. В разветвленных корпоративных сетях типа сети Интернет сложной проблемой является исключение навязывания ложной ключевой информации при взаимном обмене пользователями своими открытыми ключами.

Классические протоколы распределения ключей предписывают предварительно рассылать сформированные самими пользователями или центром формирования ключей (ЦФК) ключи по защищенным от злоумышленников каналам связи или использовать  центр распределения ключей (ЦРК), который заверяет своей цифровой подписью взаимно рассылаемые открытые ключи пользователей. В сетях, подобных Интернет, это невозможно. В корпоративной сети разнородных пользователей нет ЦРК, которому доверяли бы все корреспонденты сети. С другой стороны, разделенные тысячами километров корреспонденты сети, как правило, не имеют возможности обменяться ключами при личной встрече или разослать их через доверенных курьеров. В PGP подлинность передаваемых по сети открытых ключей проверки цифровой подписи и открытых ключей шифрования корреспондентов подтверждается цифровой подписью известных в сети корреспондентов (принцип рекомендательных писем).

Поэтому подлинность полученных по каналу связи открытых ключей нового корреспондента подтверждает цифровая подпись того корреспондента, чей открытый ключ проверки подписи достоверно известен получателю и которому доверяет корреспондент-получатель.

Для повышения подлинности передаваемых открытых ключей в PGP рекомендуется заверять их подписями нескольких корреспон дентов сети, честность которых не подвергается сомнению получателями ключевой информации. При такой схеме первоначально требуется заверить свои ключи подписью доверенных лиц, выполняющих функции нотариусов, и затем доверие к открытым ключам корреспондентов-нотариусов будет распространяться на заверенные ключи. В программе PGP для шифрования передаваемых сообщений может использоваться симметричный алгоритм IDEA и несимметричный алгоритм шифрования RSA.

Достоинством первого является более высокая скорость работы (практически на два порядка быстрее алгоритма RSA) и более высокая стойкость. Положительным свойством алгоритма RSA является возможность использования открытого ключа шифрования сообщений. Алгоритм шифрования RSA целесообразно использовать, во-первых, при невозможности конфиденциальной доставки корреспонденту секретного ключа шифрования IDEA, и во-вторых, удобно шифровать открытым ключом RSA криптографические ключи IDEA, передаваемые корреспонденту сети по доступным нарушителю каналам связи. Такое совместное использование симметричной и несимметричной криптосистем позволяет объединить высокую криптостойкость и быстродействие симметричного алгоритма шифрования с удобством несимметричного алгоритма шифрования с открытыми ключами.

Одним из основных достоинств PGP является удобство управления ключами.
Программа PGP сама формирует пару ключей (открытый и конфиденциальный) для алгоритма RSA, причем для генерации ключей используются числа, статистически очень близкие к чисто случайным. Для этого пользователю предлагается набрать с клавиатуры произвольную группу символов, чтобы измерить случайные интервалы времени между нажатиями на клавиши.
Ключи каждого пользователя PGP группируются в двух файлах, называемых в PGP связками. В связке открытых ключей хранятся открытые ключи шифрования и проверки ЦП алгоритма RSA, а в связке секретных ключей – секретные ключи формирования ЦП сообщений, ключи дешифрования RSA и ключи IDEA.
Несанкционированный доступ к связкам ключей, записанных на магнитных носителях, предотвращается с помощью известного только законному пользователю пароля.

Для удобства пользования большим количеством ключей каждому ключу –связке поставлен в соответствие уникальный идентификатор, в качестве которого, как правило, используется имя корреспондента, его адрес, время формирования ключа и другие уникальные атрибуты корреспондентов сети. При шифровании сообщений в команде PGP достаточно указать только идентификатор корреспондента-получателя, по которому программа PGP выберет из связки необходимый криптографический ключ. Еще проще управление ключами дешифрования сообщений и проверки ЦП: в передаваемом сообщении  Указывается идентификатор корреспондента-отправителя, по которому на приеме PGP автоматически извлекает из связки ключей корреспондента-получателя  необходимый ключ. Авторы PGP предусмотрели процедуру оповещения корреспондентов о компрометации своих ключей. В этом случае скомпрометированный ключ стирается из их ключевых связок.

**Однонаправленная функция с потайным ходом на основе алгебраических уравнений по модулю 2**

Для построения ОНФ с потайным ходом заманчиво использовать известные NP-сложные задач.

Известна теорема о том, что решение алгебраических уравнений по модулю 2 в общем случае является NP сложных задач. Применим эту теорему для построения ОНФ пх следующего вида:

Пусть вектор сообщения состоит из двух частей длиной по n бит каждая M=(M0, M1), где M0, M1 есть левая и правая части вектора сообщения соответственно; вектор ключа К определим, как совокупнось подвектора К=$\{K_1, K_2,\ldots, K_{d-1}\}$, каждый подвектор $K_i$, i=1,…,d-1 включает n бит из вектора ключа К. $K_i=(k_{i1},k_{i2},\ldots,k_{in})$, где $k_{ij}$Є K, j=1, 2,…,n.

Над вектором M d раз циклически выполним операции вида $Mi=M_{i-2} + f (K_{i-1}, M_{i-1})$, $2\le i\le d$, где f есть некоторое фиксированное нелинейное преобразование, и все действия выполняются по mod 2. Значение ОНФ f от аргументов M и К равно $C=f (M, K)=(M_{d-1}, M_d)$

Для законного получателя, знающего К, ОНФ обратима. Получатель рекурсивно восстанавливает входной вектор М из полученной криптограммы $C=(M_{d-1},M_d)$ по правилу:

$M_{d-2}= M_d + f(K_{d-1}, M_{d-1})$

$M_{d-3}= M_{d-1} + f(K_{d-2}, M_{d-2})$ до получения сообщения $M=(M_0, M_1)$.

Для данной ОНФ с пх обращение функции при знании ключа К (К - параметр потайного хода) является вычислительно простой задачей, как и прямая задача. Для необратимости данной функции при неизвестном ключе необходимо, чтобы преобразование f влялось нелинейным преобразованием вида: $f (k_{i1},k_{i2},\ldots,k_{im}; m_{i1},m_{i2},\ldots,m_{in}=(p_1, p_2,\ldots,p_n)$ и могло быть описано совокупностью полиномов p от тех же аргументов.

Можно показать, что при известных значениях сообщения М и криптограммы С вычисление вектора ключа К (атака со знанием открытого сообщения) сводится к решению системы алгебраических уравнений с неизвестными $К_1, К_2,\ldots,К_{d-1}$

Это-NPсложная задача. Пример. Пусть n=3,

$f(K_i, M_i)=f(k_{i1}, k_{i2}, k_{i3}; m_{i1}, m_{i2}, m_{i3})=$

$=(k_{i1}k_{i2}m_{i1}m_{i2}, k_{i2}k_{i3}m_{i1}m_{i3}, (k_{i1}\oplus k_{i2})m_{i1}m_{i3})$

Пусть вектор ключа К состоит из 8 бит вида К=(к1,к2,…,к8)

и сформируем совокупность подвекторов ключа по правилу:

$К_1=(к_2,к_4,к_6)$, $К_2=(к_1,к_2,к_7)$, $К_3=(к_3,к_5,к_8)$, $К_4=(к_6,к_7,к_8)$

Пусть вектор сообщения М=(101111), из него получим левую и правую части $M_0=(101)$ и $M_1=(111)$.

Выполним ОН преобразования вида $M_2 = M_0+f(K_1,M_1)=$

$=(101)\oplus(k_2k_4,k_4k_6,k_2\oplus k_4)=(1+ k_2k_4,k_4k_6,1\oplus k_2\oplus k_4)$, $M_3=M_1\oplus f(k_1,k_2,k_7,M_2)$

и т.д. Выполнив две итерации данной ОНФ получим криптограмму длиной 2n бит $C=(M_2,M_3)$. Данная функция вычисляется в прямом направлении и обращается при знании ключа, но при нелинейной f и неизвестном ключе, ее обращение вычислительно сложно.

Рассмотренный принцип построения ОНФ с пх используется при построении блочных систем шифрования (класс блочных шифров Фейстеля), к которому принадлежит DES и согласно ГОСТ 28147-89.

К настоящему времени предложено большое количество ОНФ с пх, построенных на основе вычислительно сложных математических задач. Наиболее часто используется сложность решения следующих теоретико-числовых задач:

-отыскание дискретного логарифма элемента в большом конечном поле или группе (криптосистема открытого распространения ключей Диффи-Хэллмана, цифровая подпись Эль-Гамаля, криптосистема цифровой подписи сообщений Шнорра и др.)

-разложение больших чисел на простые множители (цифровая подпись сообщений РША, цифровая подпись Рабина и др.

-задача об укладке целочисленного ранца( класс ранцевых систем шифрования информации Меркля-Хэллмана)

-декодирование неизвестных получателю кодов Гоппы (класс систем шифрования информации Мак-Эллиса)

# КРИПТОГРАФИЧЕСКИЕ ХЭШ-ФУНКЦИИ

Криптографические хэш-функции (КХФ) в настоящее время широко используются для решения многих задач обеспечения безопасности таких как установление подлинности сообщений, аутентификация пользователей информационных систем и сетей.

Область применения постоянно увеличивается. Уникальные свойства криптографических хэш-функций позволяют использовать их для формирования шифрующих последовательностей в шифрообразуюших устройствах, для обеспечения секретности непрерывных и дискретных сообщений, для формирования случайных чисел в криптогра-фических системах и во многих других приложениях. Понятие "криптографические хзш-функции" было определено в 1979 году в работах американского математика Р. Меркля. однако еще ранее в автоматизированных системах широко использовались некриптографические хэш-функции для оптимизации размещения и поиска данных.

Хэш-функции (ХФ) произвольного вида принадлежат классу однонаправленных функций без потайного хода. Хэш-функции отличаются от прочих функций сжатием сообщений: Произвольно длинные сообщения на входе хэш-функции преобразуются в более короткие ее выходные значения, называемые значениями хэш-кода сообщения

Определение: в общем случае хэш-функцией является произвольная функция $h$, которая имеет, как минимум, два свойства:
– сжатие: функция $h$ отображает входную дискретную последовательность произвольного конечного размера в выходную последовательность фиксированной длины $n$;
– вычислительная простота определения значения хэш-функции: для заданной функции $h$ и входной последователь-ности $X$ легко вычислить значение $h(X)$. Вычислительная простота означает, что для вычисления значения хэш-функции требуется полиномиалъное число операций относительно размерности входной последовательности.

Идея хэширования сообщений заключается в следующем. Пусть множество информационных сообщений $\{X\}$ составляет словарь $S$ объемом $|S|$. Хэширующая функция $h$ отображает сообщение $X$ в его хэш-код $H$;

$$H = h(X).$$

Если различные сообщения $X \neq X^1$ отображаются в один и тот же хэш-код , $h(X)=h(X^1)$

то данная хэш-функция допускает коллизии (склеивания) сообщений. Количество коллизий для каждого хэш-кода обозначим $t_i$, где $i = 1,2, \ldots ,N$, а $N$- общее число различных хэш-кодов. Очевидно, что число различных хэш-кодов не превышает величины $2^n$ , где $n$ – двоичная длина хэш-кодов. Одной из важнейших характеристик хэш-функции является индекс хэширования (склеивания) $I(h,S)$ функции $h$ на словаре $S$:

$$I(h,S) = 1/|S| \sum_{i=1}^{N} t_i^2 - 1$$

Если $I(h,S)=0$, то коллизий не происходит, каждое сообщение хэшируется в свой, отличный от других, хэш-код.

Хэш-функции, обеспечивающие $I(h,S)=0$ , называются совершенными хэш-функциями. Однако построить совершенную хэш-функцию для достаточно большого словаря весьма сложно. Поэтому на практике обычно используют хэш-функции, индекс склеивания которых, хотя и не равен нулю, но достаточно к нему близок.

Использование некриптографических хэш-функций, например, в базах данных позволяет существенно уменьшить емкость запоминающих устройств для размещения записей (сообщений) и ускорить доступ к ним. Адресом для размещения записей служит значение кэш-кода $h(X)$, вычисленное из двоичного представления записи $X$. Если коллизии происходят, то для их разрешения используются дополнительные механизмы, например повторное хэширование сообщений.

Хэширующие функции обычно ведут себя как случайные функции: распределение хэш-кодов равновероятно и практически отсутствует статистическая зависимость между образами и соответствующими им прообразами функции. Данные свойства хэш-функций успешно используются при сжатии и распределении сообщений, но особенно эффективно "случайные" свойства хэш-функций могут быть использованы для криптографической защиты информации.

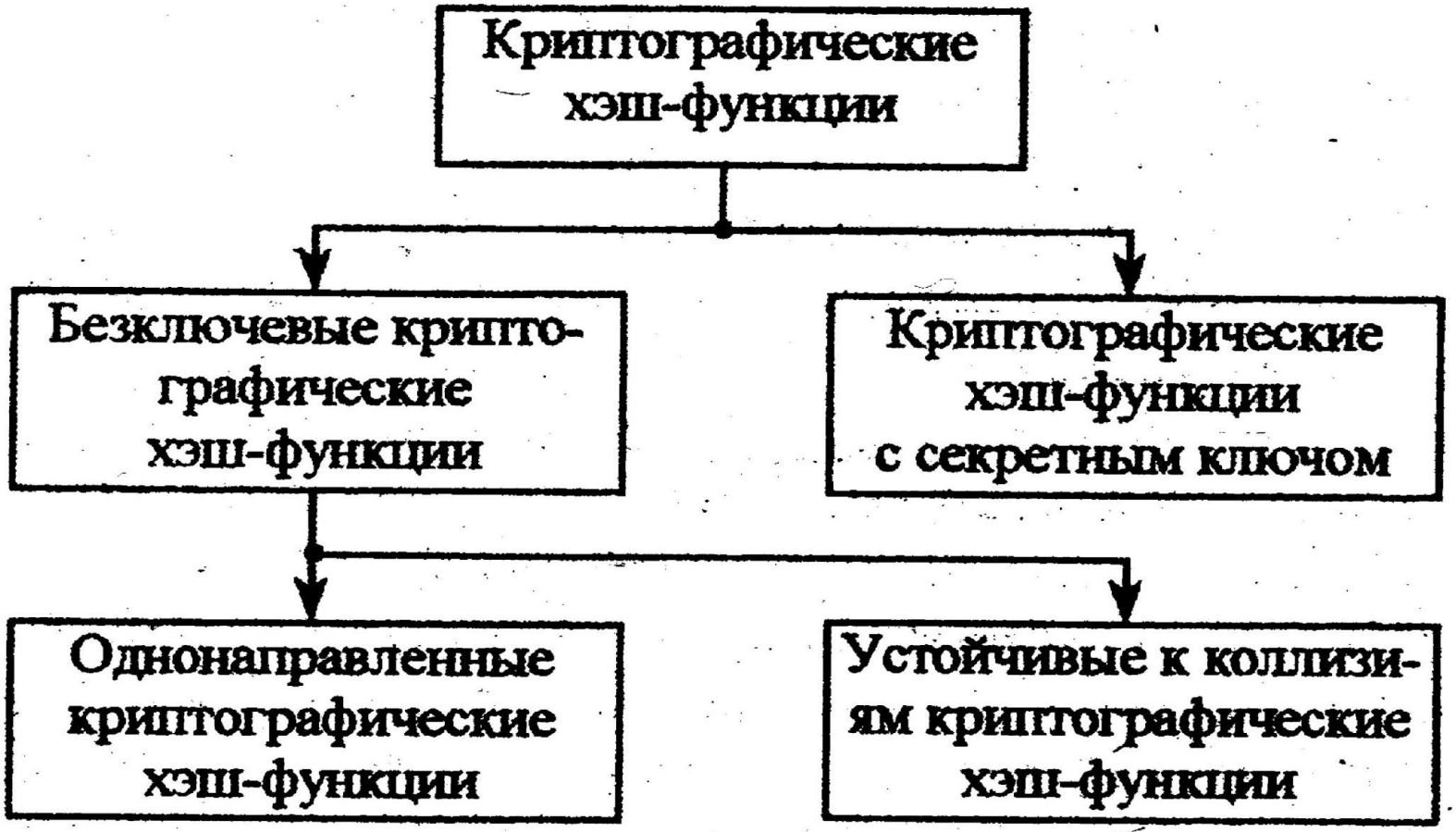
Криптографические хэш-функции, являясь подклассом хэш-функпий, должны обладать дополнительными свойствами, выполняемыми при условии, что нарушитель знает описание функции и статистические характеристики множества сообщений и множества хэш-кодов:

– стойкость к вычислению прообраза – для почти всех значений хэш-функции вычислительно невозможно отыскать любой прообраз, который хэшируется к заданному значению; – стойкость к вычислению второго прообраза – вычислительно невозможно (отыскать любой второй прообраз, который

хэшируется в такое же значение хэш-функции, как любой заданный прообраз, то есть для заданного прообраза *X* отыскать второй прообраз X1 неравный X, такой, что h(X)=h(X1);

– стойкость к образованию коллизий – вычислительно невозможно отыскать любые два различных прообраза *X* и *X'*, которые хэшируются в одинаковое значение хэш-функции, то есть выполняется h(X)=h(X1).

Первое свойство характеризует криптографические хэш-функции как однонаправленные функции: нарушитель не в состоянии вычислить сообщение *X* по его известному хзш-коду . Второе и третье свойства принципиально различаются тем. что стойкость к вычислению второго прообраза должна обеспечиваться в условиях, когда первый прообраз фиксирован для нарушителя, а стойкость к коллизиям предполагает, что он волен выбирать любые прообразы.




классификация КХФ

Если в процессе хэширования сообщений используется секретный ключ $K$, то такая функция $H=h(X, K)$ называется криптографической хэш-функцией с секретным ключом (СКХФ). Криптографические хэш-функции, не использующие секретного ключа для хэширования сообщений $H = h(X)$, называются бесключевыми криптографическими хэш-функциями (БКХФ). Бесключевые криптографические хэш-функции в зависимости от наличия перечисленных трех свойств могут быть разделены на однонаправленные КХФ и устойчивые к коллизиям КХФ.

**Бесключевые криптографические хэш-функции**

Определение 11: Бесключевая криптографическая хэш-функция называется однонаправленной криптографической хэш-функцией (ОНХФ), если она является стойкой к вычислению прообраза и стойкой к вычислению второго прообраза

Определение 12: Бесключевая криптографическая хэш-функция называется устойчивой к коллизиям криптографической хэш-функцией (УКХФ), если она является

стойкой к образованию коллизий и стойкой к вычислению второго прообраза.
Иногда используются альтернативные термины –ОНКХФ – слабые криптографические хэш-функции, а УКХФ – сильные.
Бесключевые КХФ ориентированы на обеспечение подлинности и целостности информации и предназначены для построения цифровой подписи сообщений, аутентификации пользователей и корреспондентов систем и сетей. Используются в симметричных и несимметричных системах защиты информации. Например, в несимметричных системах для исключения возможности подделки нарушителем сообщений заверяемое сообщение сначала хэшируется по БКХФ, а затем его хэш-код подписывается с использованием секретного ключа формирования ЦП сообщений.
Выбор функции зависит от условий применения и возможностей нарушителя. Если он при атаке способен выбирать сообщения для подделки, то необходимо использовать УКХФ.

Если нет – то можно использовать ОКХФ.
Стойкость ОНКХФ к вычислению прообраза может оцениваться вычислительной сложностью определения прообраза по его известному хэш-коду. Максимально достижимая стойкость к вычислению прообраза равна $O(2^n)$ операций хэширования, где n – длина хэш-кода в битах. К вычислению второго прообраза – та же. Т.е. вычисление самого сообщения, из его хэш-кода и формирование второго прообраза для фиксированного сообщения для «идеальной» ОНКХФ требует перебора всех сообщений и соответствующих им хэш-кодов. Требуется n не менее 64 бит.
Максимально достижимая стойкость УКХФ к вычислению второго прообраза - $O(2^n)$ операций хэширования, а стойкость к коллизиям - $O(2^{n/2})$ операций хэширования. Пояснить верхнюю границу стойкости можно на основе известного в математике «парадокса дня рождения». Подсчитано, что в случайно выбранной группе из 24 человек вероятность наличия хотя бы двух людей с одинаковым днем рождения

составляет не менее 0.5. Атака нарушителя на хэш-функцию с использованием «парадокса» может быть построена следующим образом: он подбирает $r_1$ истинных и $r_2$ ложных сообщений. Под ложным сообщением понимается любое сообщение, навязывание которого вместо истинного сообщения выгодно противоборствующей стороне.

Вероятность того, что хэш-код хотя бы одного ложного сообщения совпадет с хэш-кодом истинного равна: $P=1-2^{(r1r2/2n)}$

Для значений $r_1$ и $r_2$ порядка $2^{n/2.}$ Это означает, что, перебрав $r_1=r_2=2^{n/2}$ хэш-кодов, нарушитель с вероятностью 0.5 отыщет хотя бы одно ложное и коллизирующее с ним истинное, подмену которого обнаружить невозможно. В настоящее время для обеспечения вычислительной стойкости УКХФ требуется выбор n не менее 128 бит. Стойкость УКХФ выше, чем ОКНХФ. Большинство БКХФ основано на разбиении длинного сообщения на блоки фиксированной длины и их последовательной обработке КХФ одним и тем же образом.

Этот метод называется **итеративным хэшированием.** Сообщение X делится на t блоков $X_1,X_2,\ldots,X_t$ длиной по b бит.

Если длина сообщения не кратна длине блока b, она дополняется до кратной длины. Для инициализации итеративного хэширования необходимо задать: стартовый вектор хэширования $H_0$ длиной n бит. Хэширование может быть последовательно описано действиями:

$X=(X_1,\ldots.,X_i,\ \ldots,X_t)$, i=1, 2, …t

$H=f\ (X_i,\ H_{i-1})$, i=1, 2, …t

h(X)=Ht, где Hi – значение хэш-кода функции на i-той итерации хэширования, f t, где Hi – значение хэш-кода функции на i-той итерации хэширования, f отображает n битовое предыдущее состояние хэш-кода $H_{i-1}$ и b-битовый блок сообщения Xi в очередное nбитовое значение хэш-кода Hi, а значение хэш-кода и h(X) всего сообщения X определяется как значение хэш-кода на последней итерации хэширования.

Функция f называется шаговой функцией хэширования (иначе – круговая). Для обеспечения стойкости бесключевой хэш-функции важны выбор правила дополнения сообщения до требуемой длины и выбор стартового вектора хэширования.

Правило выбора стартового вектора должно исключать возможность вычислительно простого формирования коллизий.

Исследования стойкости сосредоточены на вопросе: каким условиям должна удовлетворять функция f, чтобы обеспечить стойкость. Получены два результата: 1- необходимые и достаточные условия для f ,обеспечивающей максимально достижимую стойкость ОНКХФ.

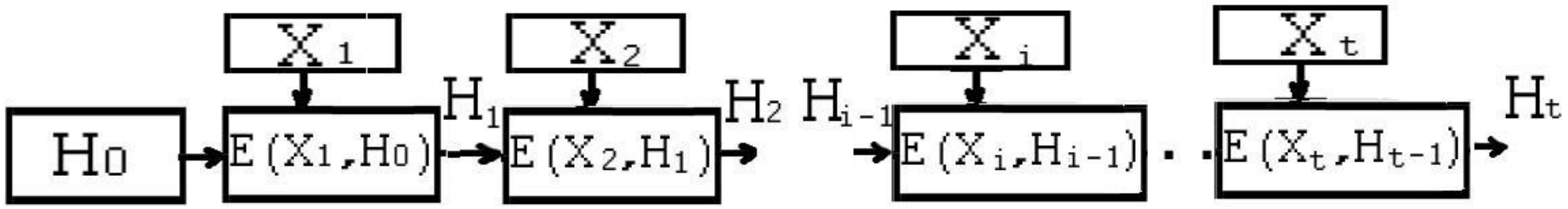
Теорема 1.Пусть дополнение включено в хэшируемое сообщение, сообщение без дополнения содержит по крайней мере 2 блока. Тогда обнаружение второго прообраза для h с фиксированным стартовым вектором хэширования требует $2^n$ вычислительных операций, если и только если для нахождения второго прообраза для шаговой функции f с произвольным выбором $H_{i-1}$ требуется $2^n$ выч. операций

Т.е. если f м.б. инвертирована менее, чем за $2^n$ операций, то для нахождения второго прообраза для h требуется не менее $2^n$ операций.

Теорема 2. Пусть дополнение однозначно и включено в хэшируемое сообщение. Если шаговая функция f устойчива к образованию коллизий, то функция h есть устойчивая к коллизиям КХФ. Обе теоремы говорят о важности выбора шаговой функции f.

**Принципы построения бесключевых криптографических хэш-функций**. Практически итеративные БКХФ делятся на два класса: функции, основанные на блочных алгоритмах и специально разработанные.

На основе **блочных алгоритмов** шифрования используется метод Уинтерница: сообщение M разбивается на блоки, по длине равные длине ключа, и на этих блоках, как на ключах, последовательно шифруется фиксированный стартовый вектор хэширования. Т.к. алгоритмы шифрования должны обеспечить стойкость к криптоанализу при атаке нарушителя, автоматически обеспечивается выч. невозможность определения прообраза по известному нарушителю хэш-коду.




Вычисление криптографической хэш-функции методом Уинтерница
Использование операции блочного хэширования E –это вычисление криптограммы Hi как функции от аргументов в виде ключа Xi и предыдущего значения хэш-кода $H_{i-1}$

$H_i = E(X_i, H_{i-1})$

Если задача поиска эквивалентных ключей шифрования для используемой операции блочного шифрования E является вычислительно сложной, то такая итерационная криптографи-ческая хэш-функция является устойчивой к образованию коллизий.

Определение: два ключа шифрования $K_1$ и $K_2$ алгоритма шифрования являются эквивалентными, если для всех сообщений X в результате их шифрования формируются одинаковые криптограммы:

$E(X, K_1) = E(X, K_2)$

Поэтому большинство известных КХФ построено на основе алгоритмов блочного шифрования.

Отдельный класс – специально разработанные хэш-функции.

К этому классу принадлежит УКХФ, называемая SHA (Security hesh algorithm), принципы построения и использования которой определены национальным стандартом США FIPS. Она предназначена для обеспечения защиты от подделки сообщений, заверяемых в криптосистеме цифровой подписи сообщений DSS в соответствии с национальным стандартом. Эта функция выполняет необратимое сжатие заверяемого сообщения многократным применением ОНФ спец.вида.

Длина хэш-кода SHA – 160 бит, поэтому достижимая оценка её стойкости к образованию коллизий составляет $2^{n/2} = 2^{80}$ операций, что считается достаточным.

В международных сетях используются УКХФ MD4 и MD5, разработанные криптографом Р.Райвестом.

Также известны ОН и УКХФ, основанные на высокой вычислительной сложности задачи факторизации большого составного числа и задачи дискретного логарифмирования в поле большой размерности соответственно. Например, ОНХФ м.б. построена итеративным использованием круговой функции $H_i = f((X_i + H_{i-1})^2 \bmod n) \oplus X_i$, где n большое составное число. Нарушитель, которому неизвестна факторизация числа n не способен из значения хэш-кода H вычислить само сообщение X. В данном классе предложено несколько функций, но скорость формирования таких КХФ существенно ниже, чем основанных на блочных алгоритмах.

Принципы построения УКХФ согласно ГОСТ Р34.11-94

Реализован в государственном стандарте России «Информационная технология Криптографическая защита информации.Функция хэширования».

Правило формирования. Хэшируемое сообщение разделяется на последовательные информационные блоки длиной по 256 бит.

Если последний блок неполный, то слева к нему дописываются нулевые символы до длины блока бит. Дополненное таким образом сообщение последовательно блок за блоком хэшируется по методу Уинтерница, включающему три этапа: генерацию несекретных ключей шифрования, шифрующее преобразование предыдущего значения хэш-кода с использованием алгоритма по ГОСТ в режиме простой замены, перемешивающее преобразование результата шифрования. На 1 этапе формируется 4 ключа по 256 бит. При хэшировании первого блока $X_1$ используется стартовый вектор хэширования $H_0$ длиной 256 бит. На 2 этапе хэш-код $H_{i-1}$ разбивается на 4 блока $h_1$-$h_4$ длиной 64 бита каждый по правилу: $H_{i-1}=h_4||h_3||h_2||h_1$, j=1, 2, 3, 4.

Каждый блок h шифруется в режиме простой замены согласно алгоритму по ГОСТ на Ki ключе шифрования. Конкатенация полученных таким образом четырех криптограмм длиной по 64 образует шифрованную последовательность длиной 256 бит: $S_i=s_4||s_3||s_2||s_1$

На третьем этапе выполняется перемешивающее преобразование Ψ шифрованной последовательности $S_i$ с учетом предыдущего значения хэш-кода $H_{i-1}$ и значения очередного хэшируемого блока сообщения $X_i$ итерационным образом:
$H_i = \Psi^{61}(H_{i-1} + \Psi(X_i \Psi^{12}(S_i)))$, где $\Psi^j$ означает j число итераций перемешивающего преобразования. Полученная величина $H_i$ является значением хэш-кода очередного информационного блока $X_i$. Итеративно обрабатывая последовательно поступающие информационные блоки пошаговой функции хэширования f формируется хэш-код всего сообщения. Хэширование последнего блока выполняется в зависимости от значений контрольной суммы и длины всего сообщения, что обеспечивает повышение стойкости данной хэш-функции к образованию коллизий и защищает от атаки нарушителя, при которой он пытается создать ложное сообщение из истинного.
Максимально достижимая оценка стойкости может составлять порядка $2^{n/2}=2^{128}$ вычислительных операций.

Криптографические хэш-функции с секретным ключом.

Определение 14: хэш-функция называется КХФ с секретным ключом, если она удовлетворяет условиям:

- простота вычисления при знании ключа: $H = h(X, K)$ вычислительно просто определяется для любого сообщения X и любого допустимого значения ключа K,
- сжатие для произвольно длинного сообщения X функция формирует хэш-код H фиксированной длины n,
-- вычислительная устойчивость: для произвольного значения $X_i$ вычислительно невозможно определение значения его хэш-кода $H_i$ без знания ключа $K_i$ даже при известном произвольно большом количестве пар $\{X_i, h(X_i, K)\}$, где $X_i \neq X_j$,
-- вычислительная необратимость относительно ключа: невозможно определние ключа K даже при известном произвольно большом количестве пар $\{X_i, h(X_i,K)\}$ вычислительно невозможно.

Криптографические хэш-функции с секретным ключом часто называют кодами аутентификации сообщений (КАС), т.к. первой областью их применения являлась аутентификация.

КХФ с секретным ключом делятся на два класса: основанные на блочных алгоритмах и специально разработанные.
Как и БКХФ КХФ с ск строятся как итеративные, использующие шаговые функции хэширования. На основе СКХФ формируется имитовставка в отечественном стандарте шифрования данных ГОСТ РФ28147-89 «Системы обработки информации.Защита криптографическая. Алгоритм криптографического преобразования.».
Принципы построения. Хэшируемое сообщение X делится на t блоков от $X_i$ до $X_t$ длиной b=64 бита. Если длина сообщения не кратна длине блока, то сообщение должно дополняться до длины, кратной длине блока. Для инициализации процесса итеративного хэширования задается нулевой стартовый вектор хэширования H0 длиной 64 бита. Хэширование сообщения X м.б.описана следующими действиями:
$X=(X_1,\ldots,X_i,\ldots,X_t)$, i=1,2,…t
$H_i=f(Xi,\oplus H_{i-1},K)$ , i=1, 2,…t, где $H_i$ –значение хэш-кода функции на i-той итерации хэширования, функция f отображает n-битовое предыдущее значение хэш-кода $H_{i-1}$ и b-битовый блок сообщения

$X_i$ в очередное n-битовое значение хэш-кода $H_i$, а значение хэш-кода h(X, K) всего сообщения X определяется как значение хэш-кода на последней итерации хэширования.

Особенностью рассматриваемой СКХФ является обязательное шифрование хэшированных сообщений. Это повышает стойкость данной хэш-функции к поиску и нарушителем коллизий и стойкость к формированию второго прообраза и оправдывает сравнительно малую длину формируемого по ГОСТ хэш-кода 16…32 бита. Это принципы построения СКХФ на основе блочных шифров в режиме сцепления блоков шифра. Такие функции формируют коды длиной 128 бит, что позволяет отказаться от обязательного шифрования самого сообщения. Стойкость их повышается при использовании вместо обратимых шаговых функций шифрования функций, не имеющих обратного преобразования. Кроме того, известны СКХФ на основе блочных шифраторов в режиме обратной связи по шифру:

$X=(X_1,\ldots,X_i,\ldots,X_t)$, $i=1,2,\ldots,t$

$H_i=f(H_{i-1},K)+X$, для $i=1,2,\ldots,t$; $h(X,K)=H_t$

Для повышения стойкости рекомендуется шифровать сформированные значения хэш-кода. Если и подлинность, и секретность сообщений обеспечиваются с использованием одного и того же блочного шифратора, криптографические ключи для обоих действий должны быть различны.

Второй класс СКХФ состоит из специально разработанных хэш-функций. Такие функции строятся из бесключевых КХФ добавлением секретного ключа  таким образом, чтобы каждый бит формируемого хэш-кода равновероятно зависел от каждого бита секретного ключа. В целом стойкость криптографических хэш-функций с секретным ключом к атакам нарушителя на подлинность сообщений оценивается выше стойкости бесключевых КХФ.

В данном классе кроме хэш-функций, обеспечивающих подлинность дискретных сообщений предложены СКХФ для сохранения секретности непрерывных и дискретных сообщений. Стойкость к попыткам нарушителя обеспечивается вычислительной устойчивостью и вычислительной необратимостью анализируемых хэш-кодов относительно сообщения и относительно ключа