С течением времени, информации становилось всё больше и больше. Алгоритмы шифрования устаревали и приходилось изобретать новые.

На данный момент есть несколько типов программ шифрования, условно их можно поделить на три группы:

1 Программы, при работе с которыми необходим только один ключ.

2 Программы, при работе с которыми необходимы несколько ключей.

3 Программы, при работе с которыми не нужны ключи.



**[1]**

Далее я расскажу об одном типе шифрования из каждой группы.

1. **Симметрическое шифрование**



**[2]**

При симметрическом шифровании данные, которые пользователь может прочитать, конвертируются в нечитаемый набор символов. Шифровка данных происходит с помощью ключа. После успешного шифрования, данные абсолютно безопасно можно передать другому пользователю или на ресивер. Для дальнейшей работы с данными необходим ключ, который использовался изначально. Соответственно, ключ является основным инструментом для работы с зашифрованными данными. Ключ практически невозможно подобрать, однако надо понимать, что ключ не должен попадать в руки третьим лицам, тогда информация снова становится незащищена. Ключ, в современных системах обычно представляет строку данных, которые получены из надежного пароля, или из совершенно случайного источника. Ключ вводится в используемую программу шифрования и данные могут быть расшифрованы. Скремблирование данных достигается с помощью симметричного алгоритма шифрования, такие как Стандарт шифрования данных (DES), расширенный стандарт шифрования (AES), или международный алгоритм шифрования данных (IDEA).

В настоящий момент одним из самых популярных алгоритмов шифрования является AES. AES обладает достаточно примитивной защитой длина его ключа составляет 256 бит. Однако это очень криптостойкий алгоритм, поскольку вскрыть такой шифр, то есть подобрать ключ к нему, не представляется возможным. Действительно, если длина ключа составляет 256 бит, то всего существует 2256 = 1,15792·1077 различных комбинаций ключей.

Также стоит отметить, что симметричное шифрование больше подходит для безопасного хранения информации. Есть целый ряд программ, которые могут осуществлять генерацию ключа и последующее его хранение. При этом ключ лучше всего хранить на USB-флэшке или диске потому что просто запомнить его не представляется возможным. В наиболее современных алгоритмах длина ключа обычно составляет 256 бит.

Подобный алгоритм шифрования очень распространён. В России также используется алгоритм аналогичный этому. Российский алгоритм – это блочный шифр, что означает, что алгоритм шифрует 64-битные блоки, а не отдельные никак не связанные символы. Алгоритм предусматривает 32 цикла, а его ключ имеет длину в 256 бит, за счёт такой защиты, алгоритм можно считать крайне надёжным и безопасным. Конечно, шифр можно разобрать, однако при использовании даже самых мощных современных компьютеров, на поиск ключа уйдёт не менее сотни лет.

1. **Асимметрическое шифрование**



**[3]**

Аналогично симметрическому, работает асимметрический ключ, он также используется для шифровки передаваемых данных, однако, вместо того, чтобы использовать тот же ключ, что использовался для шифровки, используется абсолютно другой. Важно отметить, что ключ используемый для зашифровки, доступен всем пользователям и не представляет особой ценности, так ка было сказано выше, с его помощью невозможно подобрать «частный» ключ. Именно он используется для конечной расшифровки данных, и он необходим только получателю.

Обычно, программы в основе которых лежит асимметричное шифрование, используются для безопасной передачи данных от одного пользователя к другому с помощью интернета. Использование таких программ только для хранения данных просто не целесообразно. Очень легко привести правильный пример использования такой программы: вам необходимо получить данные в зашифрованном виде. Вы случайным образом создаёте 2 ключа и отправляете открытый ключ вашему товарищу по переписке. Ваш товарищ с помощью ключа шифрует данные, отправляет их вам. А вы в свою очередь расшифровываете полученную информацию (только с помощью второго ключа).

**[4]**

1. **Хэширование**

Программы, в основе которых лежит хэширование, особое преобразование любого объема информации, в результате которого получается некое отображение, **образ**, называемый **хэшем** (hash). Хеширование данных использует алгоритм, называемый хэш-функцией. И имеет некоторые свойства: каждый хэш обозначает одни и те же данные, при этом не представляется возможным, зная хэш, получить исходные данные, а также получать данные из хэша в одиночку. Бесполезно пытаться подобрать входную информацию, чтобы получить аналогичных хэш. Метод хэширования отличается от двух других методов шифрования тем, что после шифровки информации, невозможно получить её заново в исходном виде. Подобный метод шифрования может гарантировать безопасность, поскольку даже если злоумышленник получит доступ к хэшу, он не сможет дешифровать данные содержащиеся в послании.

Если пользователь желает максимально поднять уровень защиты, он может комбинировать все три указанных выше способа шифрования.

Каждый из трёх методов имеет свои уязвимости, но при сочетании всех методов можно добиться абсолютной защиты данных. Подобная методика очень широко распространена среди крупных организаций. Использование симметричного и асимметричного методов даже имеют своё название – «цифровой конверт». Ассиметричная шифровка позволяет быстро расшифровывать данные, когда симметрическая шифровка используется преимущественно для более надёжной и скоростной передачи ключа. Рассмотрим работу программы шифрования электронной почты PGP основана на технике "цифровой конверт". Надёжность пароля достигается с помощью хэширования. Система хранить хэш пароля, а не сам пароль, соответственно при попадании хэша в руки злоумышленника, он никак не сможет им воспользоваться. В ходе проверки, система проверит хэш входящего пароля, и увидит, если результат совпадает с тем, что хранится. Сам же пароль можно обнаружить только в некоторые моменты (смена пароля или его проверка), что безусловно делает защиту более надёжной. Хеширование также используется для проверки подлинности данных с помощью секретного ключа. Хэш генерируется с использованием данных и этого ключа. Следовательно, видны только данные и хэш, а сам ключ не передается. Таким образом, если изменения будут сделаны либо с данными, либо с хэшем, они будут легко обнаружены. В заключение можно сказать, что эти методы могут быть использованы для эффективного кодирования данных в нечитаемый формат, который может гарантировать, что они останутся безопасными. Большинство современных систем обычно используют комбинацию этих методов шифрования наряду с сильной реализацией алгоритмов для повышения безопасности.

**[5]**

В настоящее время, когда информация идёт на вес золота, новые алгоритмы и программы шифрования появляются регулярно, а специалисты очень востребованы. На рынке информационной безопасности предлагается множество средств и комплексов, которые реализуют испытанные временем криптографические алгоритмы. Все эти системы объединяет принцип «прозрачного шифрования», смысл которого состоит в том, что данные шифруются в реальном времени, при этом не являясь отдельной операцией.

**[6]**

Источники:

**[1]** [<http://ypn.ru/228/classification-of-cryptographic-algorithms/>]

**[2]** **[3]** [<https://life-prog.ru/2_89996_simmetrichnoe-i-asimmetrichnoe-shifrovanie.html>]

**[4]** [<https://compress.ru/article.aspx?id=18759>]

**[5]** [[http://cendop.bmstu.ru/userfiles/materials/annotations2017.pdf](https://users.antiplagiat.ru/go?to=UrqLVKEtpQOKEMP-SNVVE9EL9EDIlr3Sl86xo-gHMoIvRiHu04bHHao7gtqthRxxm8WhS-_swsrVfPi4-SpxWPGLpaqAFBRVvveGRPrxZKiL89AegL6E2UyjAbEDYt7owyHKGOCkmlVNojcS7V_Wlg2&next=do)]

**[6]** [<http://iteranet.ru/it-novosti/2013/08/11/kakie-algoritmy-shifrovaniya-dannyx-bolee-bezopasny/>]