**Защита программного обеспечения** — комплекс мер, направленных на защиту [программного обеспечения](https://ru.wikipedia.org/wiki/Программное_обеспечение) от несанкционированного приобретения, использования, распространения, модифицирования, изучения и воссоздания аналогов.

**Защита от несанкционированного использования программ** — система мер, направленных на противодействие нелегальному использованию [программного обеспечения](https://ru.wikipedia.org/wiki/Программное_обеспечение). При защите могут применяться организационные, юридические, программные и программно-аппаратные средства.

**Защита от копирования** к программному обеспечению применяется редко, в связи с необходимостью его распространения и установки на компьютеры пользователей. Однако, от копирования может защищаться лицензия на приложение (при распространении на физическом носителе) или его отдельные алгоритмы.

## Технические средства защиты

См. также: [Технические средства защиты авторских прав](https://ru.wikipedia.org/wiki/Технические_средства_защиты_авторских_прав)

Методы можно классифицировать ***по способу распространения*** защищаемого программного обеспечения и ***типу носителя лицензии***.

### Локальная программная защита

Требование ввода серийного номера (ключа) при установке/запуске. История этого метода началась тогда, когда приложения распространялись только на физических носителях (к примеру, [компакт-дисках](https://ru.wikipedia.org/wiki/Компакт-диск)). На коробке с диском был напечатан серийный номер, подходящий только к данной копии программы.

С распространением сетей очевидным недостатком стала проблема распространения образов дисков и серийных номеров по сети. Поэтому в настоящий момент метод используется только в совокупности одним или более других методов (к примеру, организационных).

### Сетевая программная защита

* локальная

Сканирование [сети](https://ru.wikipedia.org/wiki/Локальная_вычислительная_сеть) исключает одновременный запуск двух программ с одним регистрационным ключом на двух компьютерах в пределах одной локальной сети.

Недостаток в том, что [брандмауэр](https://ru.wikipedia.org/wiki/Межсетевой_экран) можно настроить так, чтобы он не пропускал пакеты, принадлежащие защищённой программе. Правда, настройка брандмауэра требует некоторых пользовательских навыков. Кроме того, приложения могут взаимодействовать по сети (к примеру, при организации [сетевой игры](https://ru.wikipedia.org/wiki/Сетевая_игра)). В этом случае брандмауэр должен пропускать такой [трафик](https://ru.wikipedia.org/wiki/Сетевой_трафик).

* глобальная

Если программа работает с каким-то централизованным [сервером](https://ru.wikipedia.org/wiki/Сервер_(аппаратное_обеспечение)) и без него бесполезна (например, сервера [онлайн-игр](https://ru.wikipedia.org/wiki/MMOG), серверы обновлений [антивирусов](https://ru.wikipedia.org/wiki/Антивирусная_программа)). Она может передавать серверу свой серийный номер; если номер неправильный, сервер отказывает в услуге. Недостаток в том, что, существует возможность создать сервер, который не делает такой проверки. Например, существовал сервер *battle.da*, который по функциям был аналогичен [Battle.net](https://ru.wikipedia.org/wiki/Battle.net) (от компании Blizzard Entertainment), но пускал пользователей неавторизованных копий игр. Сейчас этот сервер закрыт, но существует немалое количество [PvPGN](https://ru.wikipedia.org/wiki/PvPGN)-серверов, которые также не проверяют регистрационные номера.

### Защита при помощи компакт-дисков

Программа может требовать оригинальный компакт-диск. В частности, такой способ применяется в [играх](https://ru.wikipedia.org/wiki/Компьютерная_игра). Стойкость таких защит невелика, ввиду широкого набора инструментов снятия образов компакт-дисков.[[1]](https://ru.wikipedia.org/wiki/Защита_программного_обеспечения#cite_note-1)

Как правило, этот способ защиты применяется для защиты программ, записанных на этом же компакт-диске, являющимся одновременно ключевым.

Для защиты от копирования используется:

* запись информации в неиспользуемых секторах;
* проверка расположения и содержимого «сбойных» секторов;
* проверка скорости чтения отдельных секторов.

Первые два метода практически бесполезны из-за возможности снятия полного образа с диска с использованием соответствующего [прикладного ПО](https://ru.wikipedia.org/wiki/Прикладное_программное_обеспечение). Третий метод считается более надежным (используется, в частности, в защите [StarForce](https://ru.wikipedia.org/wiki/StarForce)). Но существуют программы, которые могут эмулировать диски с учётом геометрии расположения данных, тем самым обходя и эту защиту. В StarForce, в числе прочих проверок, также выполняется проверка возможности записи на вставленный диск. Если она возможна, то диск считается не лицензионным. Однако, если образ будет записан на диск [CD-R](https://ru.wikipedia.org/wiki/CD-R), то указанная проверка пройдет. Существует возможность скрыть тип диска, чтобы [CD-R](https://ru.wikipedia.org/wiki/CD-R) или [CD-RW](https://ru.wikipedia.org/wiki/CD-RW) был виден как обычный [CD-ROM](https://ru.wikipedia.org/wiki/CD-ROM). Однако, в драйвер защиты может быть встроена проверка на наличие эмуляции.

В настоящее время наибольшую известность в мире имеют системы защиты от копирования [SecuROM](https://ru.wikipedia.org/wiki/SecuROM), [StarForce](https://ru.wikipedia.org/wiki/StarForce), [SafeDisc](https://ru.wikipedia.org/wiki/SafeDisc), [CD-RX](https://ru.wikipedia.org/w/index.php?title=CD-RX&action=edit&redlink=1) и [Tages](https://ru.wikipedia.org/w/index.php?title=Tages&action=edit&redlink=1).[[2]](https://ru.wikipedia.org/wiki/Защита_программного_обеспечения#cite_note-2)

Для многих программ указанный метод защиты недоступен ввиду отличного способа распространения (например, [shareware](https://ru.wikipedia.org/wiki/Shareware)-программы).

### Защита при помощи электронных ключей



Современные электронные ключи

[Электронный ключ](https://ru.wikipedia.org/wiki/Электронный_ключ) ([донгл](https://ru.wikipedia.org/wiki/Электронный_ключ)), вставленный в один из [портов](https://ru.wikipedia.org/wiki/Порт_(компьютер)) компьютера (с интерфейсом USB, LPT или COM) содержит ключевые данные, называемые также [лицензией](https://ru.wikipedia.org/wiki/Лицензия), записанные в него разработчиком

* информация для чтения/записи (в настоящий момент практически не применяется, так как после считывания ключ может быть сэмулирован)
* ключи аппаратных криптографических алгоритмов (используется наиболее часто)
* алгоритмы, созданные разработчиком программы (ставший доступным сравнительно недавно метод, в связи с появлением электронных ключей с микропроцессором, способным исполнять произвольный код; в настоящее время используется все чаще)

Достоинства защиты с использованием электронных ключей:

* Ключ можно вставлять в любой компьютер, на котором необходимо запустить программу
* Ключ не занимает/не требует наличия дисковода
* Электронный ключ умеет выполнять криптографические преобразования
* Современные ключи могут исполнять произвольный код, помещаемый в них разработчиком защиты (пример — [Guardant Code](https://ru.wikipedia.org/wiki/Guardant), [Senselock](https://ru.wikipedia.org/wiki/Senselock))

Стойкость защиты основывается на том, что ключевая информация защиты (криптографические ключи, загружаемый код) не покидает ключа в процессе работы с ним.

Основные недостатки:

* Цена (15—30 долларов за штуку)
* Необходимость доставки ключа конечному пользователю

Ранее к недостаткам можно было также отнести невысокое быстродействие ключа (в сравнении с CPU компьютера). Однако современные ключи достигают производительности в 1.25 DMIPS (пример — [HASP](https://ru.wikipedia.org/wiki/HASP),[Guardant](https://ru.wikipedia.org/wiki/Guardant)), а техника защиты с их помощью не предполагает постоянного обмена с ключом.

Существовавшие также ранее проблемы с установкой ключа на определенные аппаратные платформы в настоящий момент решены при помощи сетевых ключей (которые способны работать с одной или более копиями защищенного приложения, просто находясь с ним в одной локальной сети) и с помощью программных или аппаратных средств «проброса» USB-устройств по сети.

### Привязка к параметрам компьютера и активация

Привязка к информации о пользователе / серийным номерам компонентов его компьютера и последующая *активация программного обеспечения* в настоящий момент используется достаточно широко (пример: ОС Windows).

В процессе установки программа подсчитывает [код активации](https://ru.wikipedia.org/wiki/Код_активации) — контрольное значение, однозначно соответствующее установленным [комплектующим](https://ru.wikipedia.org/wiki/Аппаратное_обеспечение) компьютера и параметрам установленной ОС. Это значение передается разработчику программы. На его основе разработчик генерирует ключ активации, подходящий для активации приложения только на указанной машине (копирование установленных исполняемых файлов на другой компьютер приведет к неработоспособности программы).

Достоинство в том, что не требуется никакого специфического [аппаратного обеспечения](https://ru.wikipedia.org/wiki/Аппаратное_обеспечение), и программу можно распространять посредством [цифровой дистрибуции](https://ru.wikipedia.org/wiki/Цифровая_дистрибуция) (по Интернет).

Основной недостаток: если пользователь производит модернизацию компьютера (в случае привязки к [железу](https://ru.wikipedia.org/wiki/Аппаратное_обеспечение)), защита отказывает. Авторы многих программ в подобных случаях готовы дать новый регистрационный код. Например, [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) в [Windows XP](https://ru.wikipedia.org/wiki/Windows_XP) разрешает раз в 120 дней генерировать новый регистрационный код (но в исключительных случаях, позвонив в службу активации, можно получить новый код и после окончания этого срока).

В качестве привязки используются, в основном, серийный номер [BIOS](https://ru.wikipedia.org/wiki/BIOS) материнской платы, серийный номер винчестера. В целях сокрытия от пользователя данные о защите могут располагаться в неразмеченной области жесткого диска.

До недавнего времени такие защиты разрабатывались и внедрялись разработчиками самого программного продукта. Однако сейчас существуют SDK для работы с программными ключами, например [HASP](https://ru.wikipedia.org/wiki/HASP) SL от компании Aладдин Р. Д. Также все большее распространение получают сервисы, предлагающие одновременно функционал «навесной» защиты и сервера активации/лицензирования (пример — [Guardant Online](https://ru.wikipedia.org/w/index.php?title=Guardant_Online&action=edit&redlink=1), [Protect online](https://ru.wikipedia.org/w/index.php?title=Protect_online&action=edit&redlink=1)).

### Защита программ от копирования путём переноса их в [онлайн](https://ru.wikipedia.org/wiki/Онлайн)

Другим направлением защиты программ является использование подхода [SaaS](https://ru.wikipedia.org/wiki/SaaS), то есть предоставление функционала этих программ (всего или части), как сервиса. При этом код программы расположен и исполняется на сервере, доступном в глобальной сети. Доступ к нему осуществляется по принципу [тонкого клиента](https://ru.wikipedia.org/wiki/Тонкий_клиент). Это один из немногих случаев, когда реализуется защита от копирования.

Код исполняется на «доверенной» стороне, откуда не может быть скопирован.

Однако, и здесь возникает ряд проблем, связанных с безопасностью:

* стойкость такой защиты зависит, прежде всего, от защищенности серверов, на которых он исполняется (речь идет о Интернет-безопасности)
* важно обеспечение конфиденциальности запросов, аутентификации пользователей, целостности ресурса (возможности «горячего» резервирования), и доступности решения в целом

Возникают также вопросы доверия сервису (в том числе правовые), так как ему фактически «в открытом виде» передаются как само ПО, так и данные, которые оно обрабатывает (к примеру, [персональные данные](https://ru.wikipedia.org/wiki/Персональные_данные) пользователей).

### Защита кода от анализа

Можно выделить здесь отдельно средства защиты непосредственно кода приложения от анализа и использования в других программах. В частности, применяются [обфускаторы](https://ru.wikipedia.org/wiki/Обфускация) — программы нужны для запутывания кода с целью защиты от его анализа, модификации и несанкционированного использования.

### Защита программного обеспечения на мобильных платформах

Основная статья: [**Мобильная игра#Защита от копирования**](https://ru.wikipedia.org/wiki/Мобильная_игра#.D0.97.D0.B0.D1.89.D0.B8.D1.82.D0.B0_.D0.BE.D1.82_.D0.BA.D0.BE.D0.BF.D0.B8.D1.80.D0.BE.D0.B2.D0.B0.D0.BD.D0.B8.D1.8F)

Способы защиты программного обеспечения для мобильных платформ от копирования обычно основываются на невозможности рядового пользователя считывать/изменять хранящиеся в [ППЗУ](https://ru.wikipedia.org/wiki/ППЗУ) аппарата данные. Может также использоваться активация программного обеспечения.