A felhasználó (*initiator*) egy Tor klienst futtat a gépén, amely csatlakozik a Tor hálózathoz. Időnként megújítja az egész kimenő útvonalat *(circuit)*. A hálózatban részt vevő tagok opcionálisan konfigurálhatják a Tor kliensüket úgynevezett *node*-nak, vagy Tor-*exit*-nek.

A *node* engedélyezi, hogy rajta keresztül csomagok fussanak, de nem engedélyezi a hálózatból való kilépést a hagyományos internetre, míg a Tor-*exit* (Tor kilépő) ezt is engedélyezi.

A Tor hálózat minden tagja titkosítva kommunikál egymással 128 bites szimmetrikus kulcsolású kódokat használva, melyeket egy aszimmetrikus kulcsolású ún. *handshake* (kézfogás) után hoztak létre.

A kezdeményező minden csomagra először rátesz egy véletlenszerű mesterkulcsot (egy kódot, melyet csak ő és végül a Tor-*exit* gép ismer), majd a mesterkulccsal titkosított csomag és a mesterkulcs három részre tagolódik és minden csomagrészlet kap külön-külön egy-egy első szintű kulcsot. Az első szintű kulcs az a kulcs, amelyet az első szintű *node*-okkal (a közvetlen kapcsolatokkal) hoztunk létre a *handshake* után.

Mind a szintek, mind az útvonal megválasztása a Tor hálózatban véletlenszerűen zajlik, valamint véletlenszerű időközönként az *initiator* még akkor is megújítja az egész útvonalat, ha egyébként a *circuit* gyors és stabil.

Egy-egy ösvény *(path)*, amelyen valamelyik csomagrészletünk halad, megszakadhat, ha valamelyik *node* vagy a Tor-*exit* időközben elérhetetlenné válik. Ekkor az az ösvény újra kiépül, immár más, véletlenszerűen kiválasztott *node*-okkal (és esetleg Tor-*exittel*).

A csomagrészlet és a mesterkulcsrészlet az első szintű *node*-okhoz került, ezek ráteszik a saját (második szintű) kulcsukat és továbbküldik a csomagrészleteket a harmadik szintű *node*-nak.

A Tor hálózat *path*-ai három szintűek, illetve három részre tagolódnak (tehát három különböző *path*-on haladnak), így az *initiator*ral és a Tor-*exittel* együtt összesen 11 számítógép vesz részt egy felépült *circuitban*, 3x3 *node* közreműködésével.

A harmadik szintű *node*-ok végül elküldik a csomagrészleteket a Tor-*exitnek*, ami visszafejti a csomagok részleteit visszafele, először a harmadik szint, majd a második, majd az első szintű csomagrészletekig.

Minden *node* függetlenül attól, hogy hányadik a sorban, elküldi a soron következőnek a kapott csomagrészleteket a "kapott kulccsal" (előző szintű kulccsal). Így minden *node* visszafele megismerheti a legelső csomag*részlet* titkosított változatát, beleértve a Tor-*exitet* is, viszont a *node*-ok annak csak a titkosított változatát ismerhetik, mivel minden *node*-nál csupán a mesterkulcsunk harmadrésze és a csomagok mesterkulccsal titkosított változatának harmadrésze van.

Ezután a Tor-*exit* a három különböző irányból megkapott mesterkulcsot összeillesztve, azzal visszafejti az eredeti csomagot. Végül pedig immár a hagyományos, „kódolatlan interneten” keresztül továbbítja az eredeti célnak.

#### Exit node eavesdropping

In September 2007, Dan Egerstad, a Swedish security consultant, revealed that he had intercepted usernames and passwords for e-mail accounts by operating and monitoring Tor exit nodes. As Tor cannot encrypt the traffic between an exit node and the target server, any exit node is in a position to capture traffic passing through it that does not use end-to-end encryption such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS). While this may not inherently breach the anonymity of the source, traffic intercepted in this way by self-selected third parties can expose information about the source in either or both of payload and protocol data. Furthermore, Egerstad is circumspect about the possible subversion of Tor by intelligence agencies:

"If you actually look in to where these Tor nodes are hosted and how big they are, some of these nodes cost thousands of dollars each month just to host because they're using lots of bandwidth, they're heavy-duty servers and so on. Who would pay for this and be anonymous?"

In October 2011, a research team from ESIEA claimed to have discovered a way to compromise the Tor network by decrypting communication passing over it. The technique they describe requires creating a map of Tor network nodes, controlling one third of them, and then acquiring their encryption keys and algorithm seeds. Then, using these known keys and seeds, they claim the ability to decrypt two encryption layers out of three. They claim to break the third key by a statistical-based attack. In order to redirect Tor traffic to the nodes they controlled, they used a denial-of-service attack. A response to this claim has been published on the official Tor Blog stating that these rumours of Tor's compromise are greatly exaggerated.

### Tor exit node block

Operators of Internet sites have the ability to prevent traffic from Tor exit nodes or to offer reduced functionality to Tor users. For example, it is not generally possible to edit Wikipedia when using Tor or when using an IP address that also is used by a Tor exit node, due to the use of the [TorBlock](https://www.mediawiki.org/wiki/Extension:TorBlock) MediaWiki extension, unless an exemption is obtained. The BBC blocks the IP addresses of all known Tor guards and exit nodes from its [iPlayer](https://en.wikipedia.org/wiki/IPlayer) service - however relays and bridges are not blocked.

### Some protocols expose IP addresses

Researchers from the French Institute for Research in Computer Science and Automation (INRIA) showed that the Tor dissimulation technique in [BitTorrent](https://en.wikipedia.org/wiki/BitTorrent) can be bypassed by attackers controlling a Tor exit node. The study was conducted by monitoring six exit nodes for a period of 23 days. Researchers used three attack vectors:

* Inspection of BitTorrent control messages

Tracker announces and extension protocol handshakes may optionally contain client IP address. Analysis of collected data revealed that 35% and 33% of messages, respectively, contained addresses of clients.

* Hijacking trackers' responses

Due to lack of encryption or authentication in communication between tracker and peer, typical man-in-the-middle attacks allow attackers to determine peer IP addresses and even verify the distribution of content. Such attacks work when Tor is used only for tracker communication.

* Exploiting distributed hash tables (DHT)

This attack exploits the fact that distributed hash table (DHT) connections through Tor are impossible, so an attacker is able to reveal a target's IP address by looking it up in the DHT even if the target uses Tor to connect to other peers.

With this technique, researchers were able to identify other streams initiated by users, whose IP addresses were revealed.

### Sniper attack

Jansen *et al*., describe a [DDoS](https://en.wikipedia.org/wiki/DDoS) attack targeted at the Tor node software, as well as defenses against that attack and its variants. The attack works using a colluding client and server, and filling the queues of the exit node until the node runs out of memory, and hence can serve no other (genuine) clients. By attacking a significant proportion of the exit nodes this way, an attacker can degrade the network and increase the chance of targets using nodes controlled by the attacker.

### Heartbleed bug

The [Heartbleed](https://en.wikipedia.org/wiki/Heartbleed_bug) [OpenSSL](https://en.wikipedia.org/wiki/OpenSSL) bug disrupted the Tor network for several days in April 2014 while private keys were renewed. The Tor Project recommended that Tor relay operators and hidden service operators revoke and generate fresh keys after patching OpenSSL, but noted that Tor relays use two sets of keys and that Tor's multi-hop design minimizes the impact of exploiting a single relay. 586 relays later found to be susceptible to the Heartbleed bug were taken off-line as a precautionary measure.