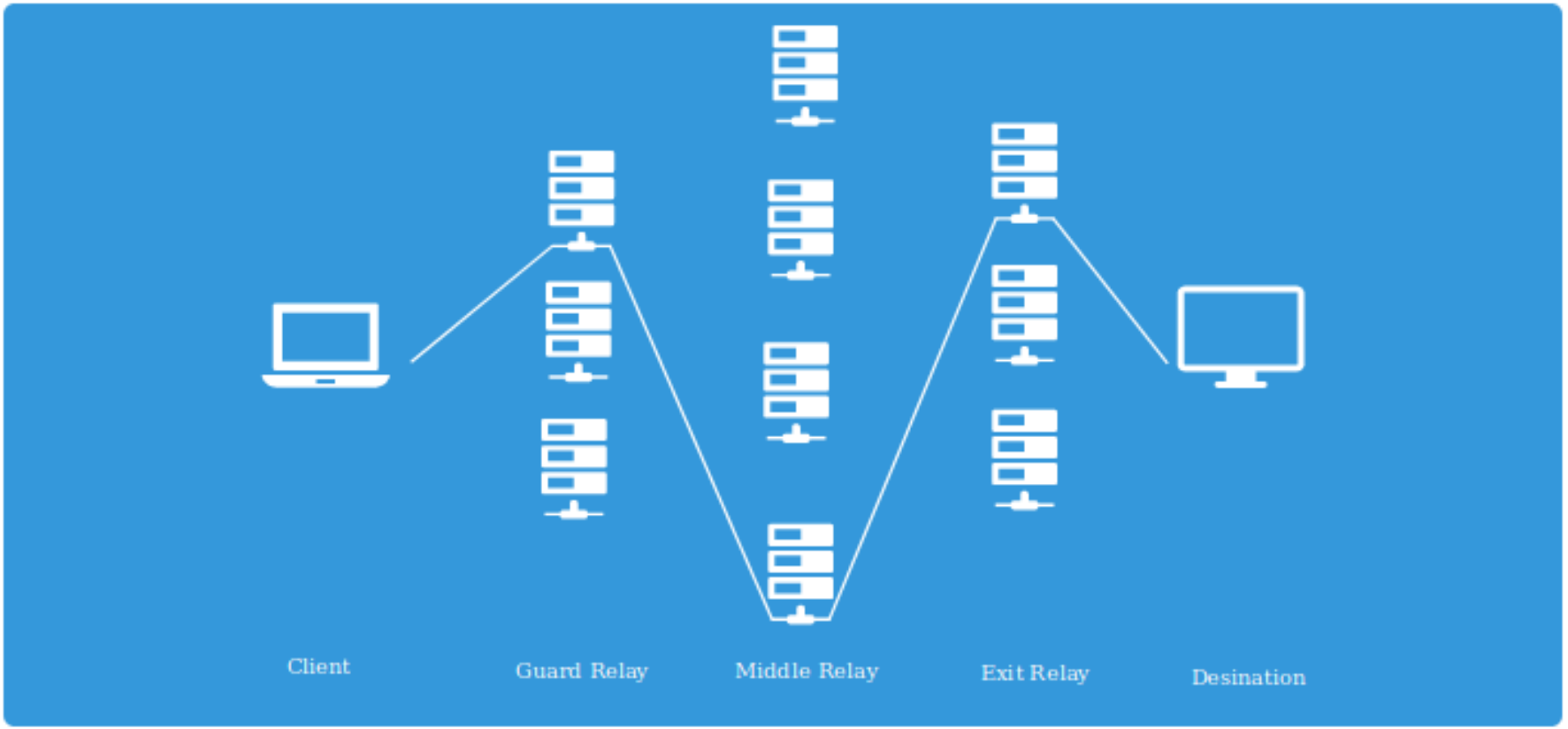
The Onion Router (Tor)

Cornel Bucurescu, Adrian Manea, Cristian Nicolae

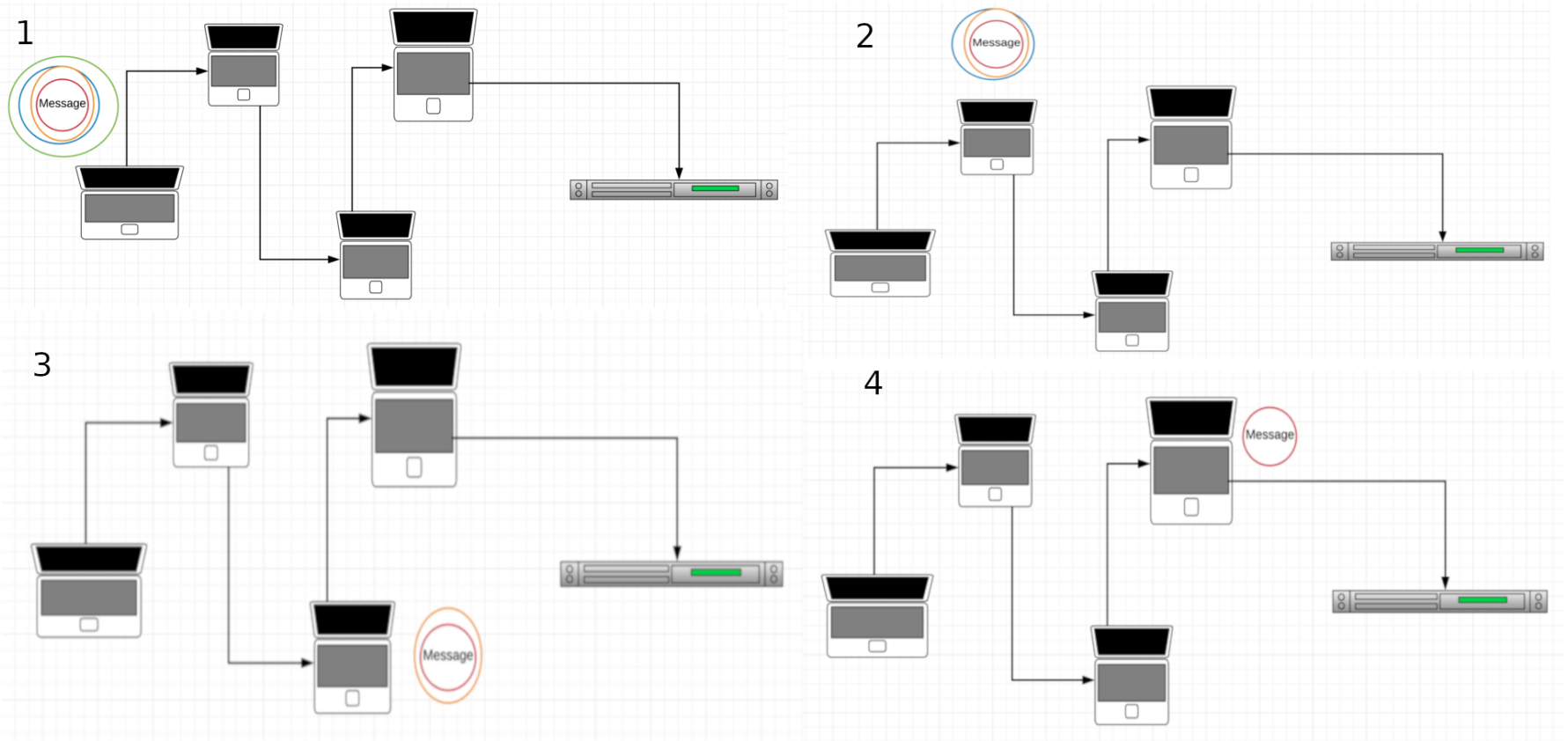
SLA, 406 & 410

Ideea principală (TL;DR)



Ilustrație: Cele 3 relee dintr-o conexiune standard [Wright, 2015]

Ideea principală (TL;DR)



Ilustrație: Criptarea telescopică [Skerritt, 2018]

Obiective de design (2004)

- perfect forward secrecy;

Obiective de design (2004)

- perfect forward secrecy;
- separarea filtrării de anonimitate;

Obiective de design (2004)

- perfect forward secrecy;
- separarea filtrării de anonimitate;
- TCP streams multiplexing;

Obiective de design (2004)

- perfect forward secrecy;
- separarea filtrării de anonimitate;
- TCP streams multiplexing;
- topologie “leaky pipe”;

Obiective de design (2004)

- perfect forward secrecy;
- separarea filtrării de anonimitate;
- TCP streams multiplexing;
- topologie “leaky pipe”;
- autorități directoare;

Obiective de design (2004)

- perfect forward secrecy;
- separarea filtrării de anonimitate;
- TCP streams multiplexing;
- topologie “leaky pipe”;
- autorități directoare;
- servicii ascunse.

Non-obiective (limitări admise)

- nu servește drept conexiune peer-to-peer (vulnerabilă!);

Non-obiective (limitări admise)

- nu servește drept conexiune peer-to-peer (vulnerabilă!);
- nu protejează împotriva atacurilor end-to-end (timing, intersection);

Non-obiective (limitări admise)

- nu servește drept conexiune peer-to-peer (vulnerabilă!);
- nu protejează împotriva atacurilor end-to-end (timing, intersection);
- nu normalizează protocoalele;

Non-obiective (limitări admise)

- nu servește drept conexiune peer-to-peer (vulnerabilă!);
- nu protejează împotriva atacurilor end-to-end (timing, intersection);
- nu normalizează protocoalele;
- nu ascunde traficul ($\neq$ *steganografic*).

Funcționare

Modelul este de “overlay network”.

Funcționare

Modelul este de “overlay network”.

OR au chei de identitate (IDK) și cheie onion (OK).

Funcționare

Modelul este de “overlay network”.

OR au chei de identitate (IDK) și cheie onion (OK).

IDK semnează certificate TLS și directoarele.

Funcționare

Modelul este de “overlay network”.

OR au chei de identitate (IDK) și cheie onion (OK).

IDK semnează certificate TLS și directoarele.

OK decriptează cererile, stabilește circuite și negociază cheile efemere.

Funcționare

Modelul este de “overlay network”.

OR au chei de identitate (IDK) și cheie onion (OK).

IDK semnează certificate TLS și directoarele.

OK decriptează cererile, stabilește circuite și negociază cheile efemere.

Comunicarea se face prin TLS cu chei efemere.

Funcționare

Modelul este de “overlay network”.

OR au chei de identitate (IDK) și cheie onion (OK).

IDK semnează certificate TLS și directoarele.

OK decriptează cererile, stabilește circuite și negociază cheile efemere.

Comunicarea se face prin TLS cu chei efemere.

Traficul circulă în *celule de cîte 512 bytes*, cu *header* și *payload*.

Construcția unui circuit Alice $\longrightarrow$ Bob

Alice $\xrightarrow{\texttt{create}}$ Bob și alege `circID` C_{AB}.

Construcția unui circuit Alice $\longrightarrow$ Bob

Alice $\xrightarrow{\texttt{create}}$ Bob și alege `circID` C_{AB}.

Alice $\xrightarrow{\texttt{OK}=g^x}$ Bob.

Construcția unui circuit Alice $\longrightarrow$ Bob

Alice $\xrightarrow{\texttt{create}}$ Bob și alege `circID` C_{AB}.

Alice $\xrightarrow{\texttt{OK}=g^x}$ Bob.

Bob $\xrightarrow{\texttt{created}=g^y,\quad hash(K=g^{xy})}$ Alice.

Construcția unui circuit Alice $\longrightarrow$ Bob

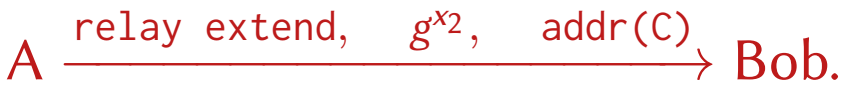
Alice $\xrightarrow{\texttt{create}}$ Bob și alege `circID` C_{AB}.

Alice $\xrightarrow{\texttt{OK}=g^x}$ Bob.

Bob $\xrightarrow{\texttt{created}=g^y,\quad hash(K=g^{xy})}$ Alice.

Acum, Alice și Bob pot comunica folosind cheia K.

Extinderea circuitului Alice → Bob → Carol



Extinderea circuitului Alice → Bob → Carol

A $\xrightarrow{\texttt{relay extend}, \quad g^{x_2}, \quad \texttt{addr(C)}}$ Bob.

Bob $\xrightarrow{\texttt{create}}$ Carol && copy(g^{x_2}) && alege `circID` C_{BC}.

Extinderea circuitului Alice → Bob → Carol

A $\xrightarrow{\texttt{relay extend},\quad g^{x_2},\quad \texttt{addr(C)}}$ Bob.

Bob $\xrightarrow{\texttt{create}}$ Carol && copy(g^{x_2}) && alege `circID` C_{BC}.

Carol $\xrightarrow{\texttt{created}=g^{y_2}}$ Bob $\xrightarrow{\texttt{payload}}$ Carol.

Extinderea circuitului Alice → Bob → Carol

A $\xrightarrow{\texttt{relay extend}, \quad g^{x_2}, \quad \texttt{addr(C)}}$ Bob.

Bob $\xrightarrow{\texttt{create}}$ Carol && copy(g^{x_2}) && alege `circID` C_{BC}.

Carol $\xrightarrow{\texttt{created}=g^{y_2}}$ Bob $\xrightarrow{\texttt{payload}}$ Carol.

Bob $\xrightarrow{\texttt{relay extended}}$ Alice.

Extinderea circuitului Alice → Bob → Carol

A $\xrightarrow{\texttt{relay extend}, \quad g^{x_2}, \quad \texttt{addr(C)}}$ Bob.

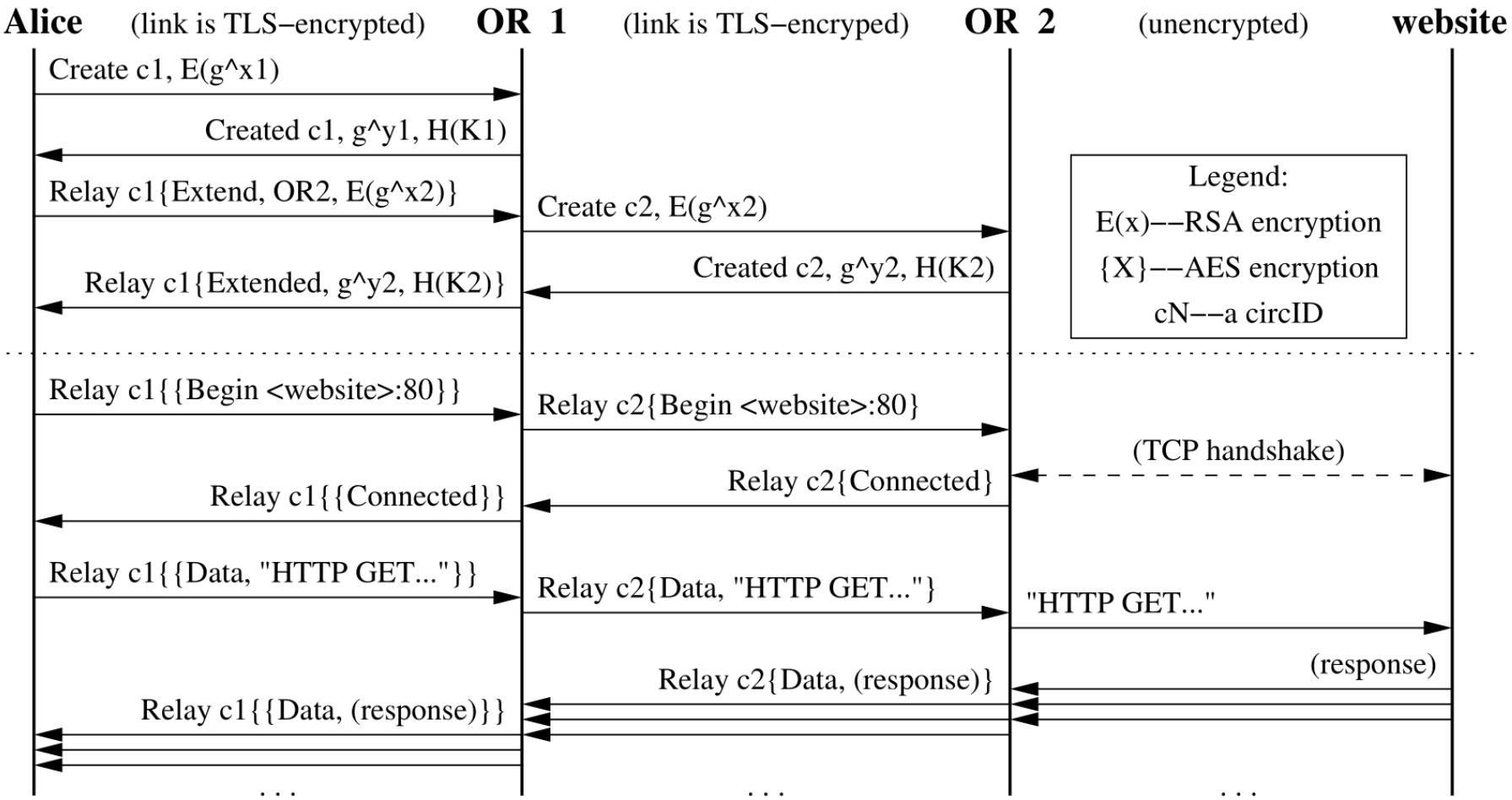
Bob $\xrightarrow{\texttt{create}}$ Carol && copy(g^{x_2}) && alege `circID` C_{BC}.

Carol $\xrightarrow{\texttt{created}=g^{y_2}}$ Bob $\xrightarrow{\texttt{payload}}$ Carol.

Bob $\xrightarrow{\texttt{relay extended}}$ Alice.

Acum, se poate comunica A—C cu cheia $K_2 = g^{x_2 y_2}$.

Circuit cu 3 relee



Ilustrație: Celule schimbate la inițierea unui circuit ([Dingledine et al., 2004], §4.1

Servicii ascunse ([hash].onion)

Serverul ascuns creează o pereche de chei pentru criptare asimetrică și cere unui OR să devină punct de intrare.

Servicii ascunse ([hash].onion)

Serverul ascuns creează o pereche de chei pentru criptare asimetrică și cere unui OR să devină punct de intrare.

După acceptare, serverul pune într-un *tabel de hash-uri distribuit* IP-urile punctelor de introducere acceptate.

Servicii ascunse ([hash].onion)

Serverul ascuns creează o pereche de chei pentru criptare asimetrică și cere unui OR să devină punct de intrare.

După acceptare, serverul pune într-un *tabel de hash-uri distribuit* IP-urile punctelor de introducere acceptate.

Clientul primește adresa de tip [hash].onion (cheia publică), deci știe IP-urile cu care poate intra în Tor.

Servicii ascunse ([hash].onion)

Serverul ascuns creează o pereche de chei pentru criptare asimetrică și cere unui OR să devină punct de intrare.

După acceptare, serverul pune într-un *tabel de hash-uri distribuit* IP-urile punctelor de introducere acceptate.

Clientul primește adresa de tip [hash].onion (cheia publică), deci știe IP-urile cu care poate intra în Tor.

Clientul creează o celulă de introducere, cu adresa nodului ales pentru acces.

Servicii ascunse ([hash].onion)

Serverul ascuns creează o pereche de chei pentru criptare asimetrică și cere unui OR să devină punct de intrare.

După acceptare, serverul pune într-un *tabel de hash-uri distribuit* IP-urile punctelor de introducere acceptate.

Clientul primește adresa de tip [hash].onion (cheia publică), deci știe IP-urile cu care poate intra în Tor.

Clientul creează o celulă de introducere, cu adresa nodului ales pentru acces.

Nodul decriptează celula de introducere cu cheia privată și acceptă clientul.

Servicii ascunse ([hash].onion)

Serverul ascuns creează o pereche de chei pentru criptare asimetrică și cere unui OR să devină punct de intrare.

După acceptare, serverul pune într-un *tabel de hash-uri distribuit* IP-urile punctelor de introducere acceptate.

Clientul primește adresa de tip [hash].onion (cheia publică), deci știe IP-urile cu care poate intra în Tor.

Clientul creează o celulă de introducere, cu adresa nodului ales pentru acces.

Nodul decriptează celula de introducere cu cheia privată și acceptă clientul.

Comunicarea se face cu un nod intermediar, deci conexiunea cu serverul ascuns are cel puțin 6 noduri.

Autorități directoare (AD)

Există cîteva autorități directoare = noduri de încredere.

Autorități directoare (AD)

Există cîteva autorități directoare = noduri de încredere.

AD votează și actualizează statusul rețelei în *consens*:

Autorități directoare (AD)

Există cîteva autorități directoare = noduri de încredere.
AD votează și actualizează statusul rețelei în *consens*:

- fac o listă de relee cunoscute;

Autorități directoare (AD)

Există cîteva autorități directoare = noduri de încredere.
AD votează și actualizează statusul rețelei în *consens*:

- fac o listă de relee cunoscute;
- calculează alți parametri necesari (țara, lățimea de bandă);

Autorități directoare (AD)

Există cîteva autorități directoare = noduri de încredere.
AD votează și actualizează statusul rețelei în *consens*:

- fac o listă de relee cunoscute;
- calculează alți parametri necesari (țara, lățimea de bandă);
- transmit cele de mai sus către celelalte AD, ca status;

Autorități directoare (AD)

Există cîteva autorități directoare = noduri de încredere.
AD votează și actualizează statusul rețelei în *consens*:

- fac o listă de relee cunoscute;
- calculează alți parametri necesari (țara, lățimea de bandă);
- transmit cele de mai sus către celelalte AD, ca status;
- rezultă un „status mediat“;

Autorități directoare (AD)

Există cîteva autorități directoare = noduri de încredere.
AD votează și actualizează statusul rețelei în *consens*:

- fac o listă de relee cunoscute;
- calculează alți parametri necesari (țara, lățimea de bandă);
- transmit cele de mai sus către celelalte AD, ca status;
- rezultă un „status mediat“;
- din status $\Rightarrow$ vot, transmis între AD cu semnătură.

Autorități directoare (AD)

Există cîteva autorități directoare = noduri de încredere.
AD votează și actualizează statusul rețelei în *consens*:

- fac o listă de relee cunoscute;
- calculează alți parametri necesari (țara, lățimea de bandă);
- transmit cele de mai sus către celelalte AD, ca status;
- rezultă un „status mediat“;
- din status $\Rightarrow$ vot, transmis între AD cu semnătură.

Votul și actualizarea sînt publice.

Atacuri × apărări

Pasive:

Atacuri × apărări

Pasive:

- Observarea tiparelor de trafic × TCP streams multiplexing;

Atacuri × apărări

Pasive:

- Observarea tiparelor de trafic × TCP streams multiplexing;
- End-to-end correlation × leaky pipe topology;

Atacuri × apărări

Pasive:

- Observarea tiparelor de trafic × TCP streams multiplexing;
- End-to-end correlation × leaky pipe topology;
- Confirmation attacks, website fingerprinting × celule mici, padding.

Atacuri × apărări

Pasive:

- Observarea tiparelor de trafic × TCP streams multiplexing;
- End-to-end correlation × leaky pipe topology;
- Confirmation attacks, website fingerprinting × celule mici, padding.

Active:

Atacuri × apărări

Pasive:

- Observarea tiparelor de trafic × TCP streams multiplexing;
- End-to-end correlation × leaky pipe topology;
- Confirmation attacks, website fingerprinting × celule mici, padding.

Active:

- OR impersonation × chei efemere;

Atacuri × apărări

Pasive:

- Observarea tiparelor de trafic × TCP streams multiplexing;
- End-to-end correlation × leaky pipe topology;
- Confirmation attacks, website fingerprinting × celule mici, padding.

Active:

- OR impersonation × chei efemere;
- Legal backdoors, probleme politice sau sociale × ???;

Atacuri × apărări

Pasive:

- Observarea tiparelor de trafic × TCP streams multiplexing;
- End-to-end correlation × leaky pipe topology;
- Confirmation attacks, website fingerprinting × celule mici, padding.

Active:

- OR impersonation × chei efemere;
- Legal backdoors, probleme politice sau sociale × ???;
- Cenzură la ieșire × ???;

Atacuri × apărări

Pasive:

- Observarea tiparelor de trafic × TCP streams multiplexing;
- End-to-end correlation × leaky pipe topology;
- Confirmation attacks, website fingerprinting × celule mici, padding.

Active:

- OR impersonation × chei efemere;
- Legal backdoors, probleme politice sau sociale × ???;
- Cenzură la ieșire × ???;
- Cenzură la intrare × bridges.

Citări


Dingledine, R., Mathewson, N., and Sylverson, P. (2004).
Tor: The second generation onion router.
SSYM Proceedings of the 13th conference on USENIX Security Symbposium, 13.


Skerritt, B. (2018).
How Tor *really* works.
HackerNoon.


Tor (2019).
The Tor Project.
`https://www.torproject.org/`.


Wright, J. (2015).
How Tor Works: Part One.
blog post.