Enhancing Security and Privacy of Tor's Ecosystem by using Trusted Execution Environments

Seongmin Kim, Juhyeng Han, Jaehyeong Ha,

Taesoo Kim *, Dongsu Han

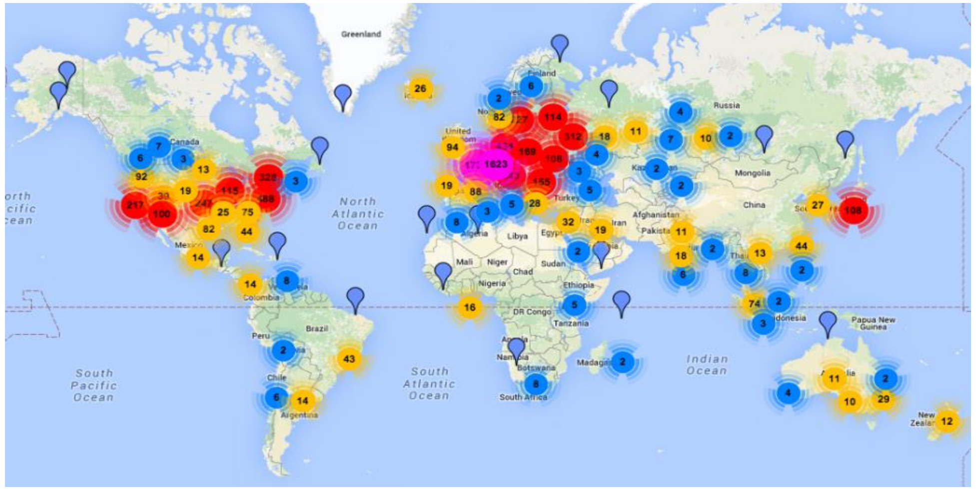
KAIST

Georgia Tech *

Tor anonymity network

- Tor: the most popular anonymity network for Internet users
  - Helps users to defend against traffic analysis and keep user's privacy (e.g., what sites you visit, IP address) [*from Tor project, www.torproject.org*]
  - Freely available as an open source
  - 1.8 million users on a daily basis

The geographic location of Tor relays *



** from Onionview, https://onionview.codeplex.com/*

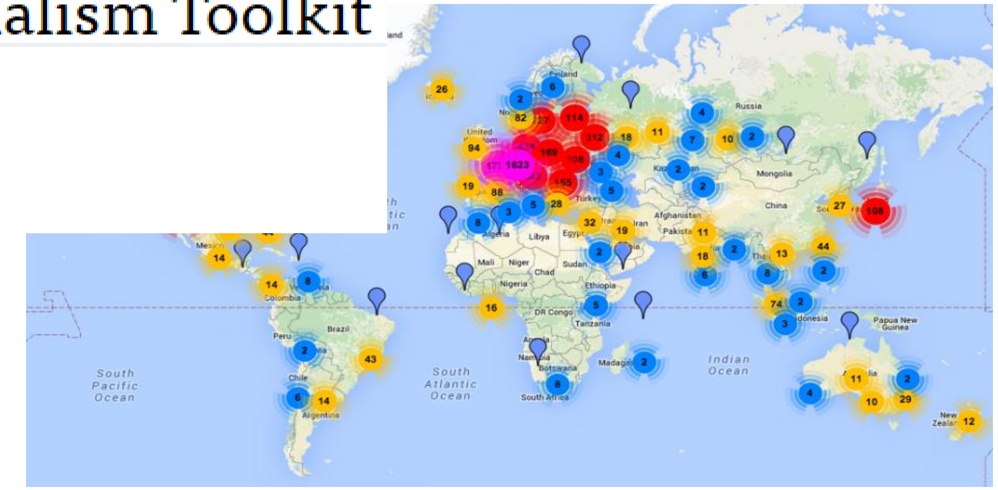
Tor anonymity network

- Tor: the most popular anonymity network for Internet users
  - Helps users to defend against traffic analysis and keep user's privacy (e.g., what sites you visit, IP address) [*from Tor project, www.torproject.org*]
  - Freely available as an open source
  - 1.8 million users on a daily basis

Tor Project Offers a Secure, Anonymous Journalism Toolkit

by Karen Reilly September 5, 2012

The geographic location of Tor relays *



** from Onionview, https://onionview.codeplex.com/*

Tor anonymity network

- Tor: the most popular anonymity network for Internet users
  – Helps users to defend against traffic analysis and keep user’s privacy (e.g., what sites you visit, IP address) [*from Tor project, www.torproject.org*]
  – Freely available as an open source
  – 1.8 million users on a daily basis

Tor Project Offers a Secure, Anonymous Journalism Toolkit

by Karen Reilly September 5, 2012

The geographic location of Tor relays *

Not anonymous: attack reveals BitTorrent users on Tor network

An ingenious attack by French researchers has found a way to identify ...

THOMAS LOWENTHAL - 4/13/2011, 12:57 AM

* *from Onionview, https://onionview.codeplex.com/*

Tor anonymity network

- Tor: the most popular anonymity network for Internet users
  - Helps users to defend against traffic analysis and keep user's privacy (e.g., what sites you visit, IP address) [*from Tor project, www.torproject.org*]
  - Freely available as an open source
  - 1.8 million users on a daily basis

Tor Project Offers a Secure, Anonymous Journalism Toolkit

by Karen Reilly September 5, 2012

"One cell is enough to break Tor's anonymity"

Posted February 19th, 2009 by arma in attacks, research, tagging

The geographic location of Tor relays *

Not anonymous: attack reveals BitTorrent users on Tor network

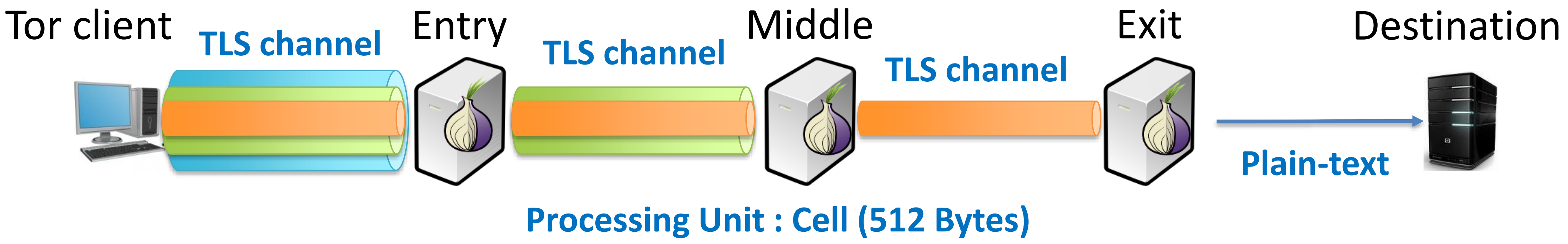
rench researchers has found a way to identify ...

:57 AM

ttps://onionview.codeplex.com/

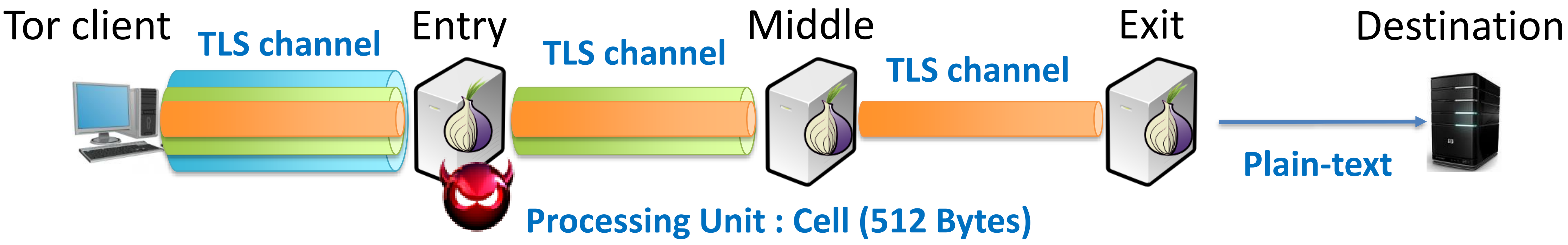
Tor network: Threat model

- 3-hop onion routing: a single Tor entity cannot know both client and server



Tor network: Threat model

- 3-hop onion routing: a single Tor entity cannot know both client and server

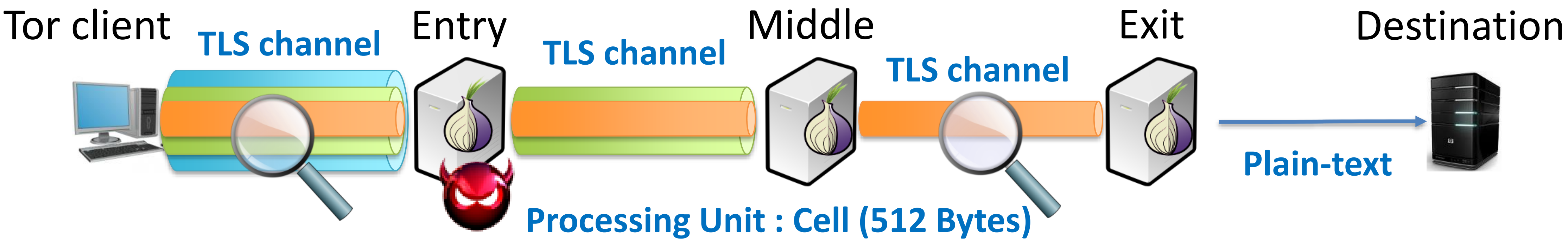


- Tor's Threat model
  - Tor is a volunteer-based network: **Tor relays are not trusted**

Can run a Tor relays of his own	**Can compromise some fraction of Tor relays**

Tor network: Threat model

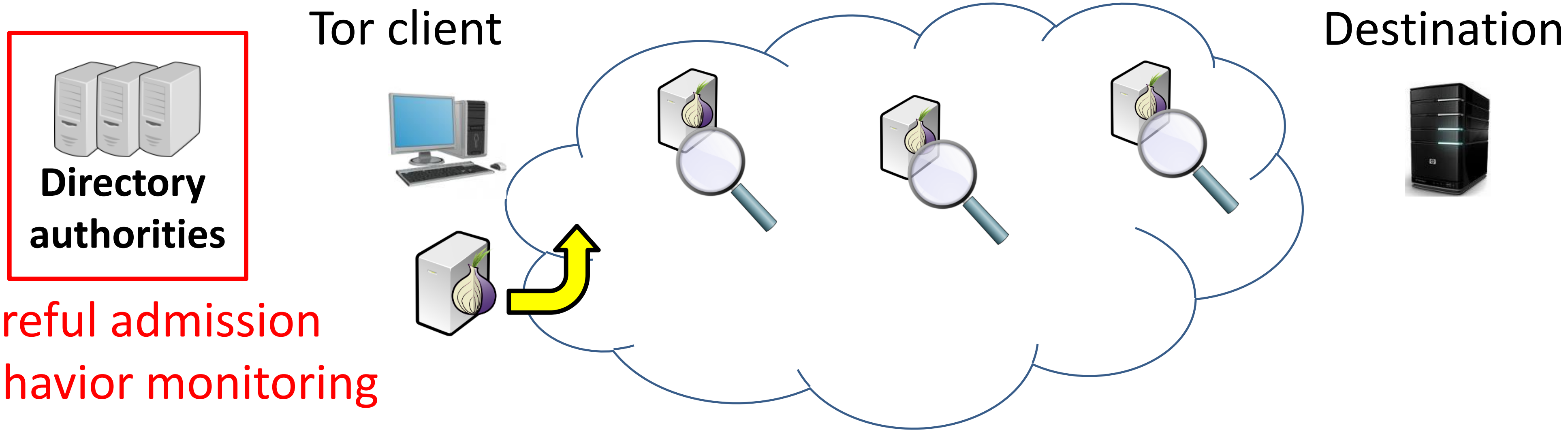
- 3-hop onion routing: a single Tor entity cannot know both client and server



- Tor's Threat model
  - Tor is a volunteer-based network: **Tor relays are not trusted**

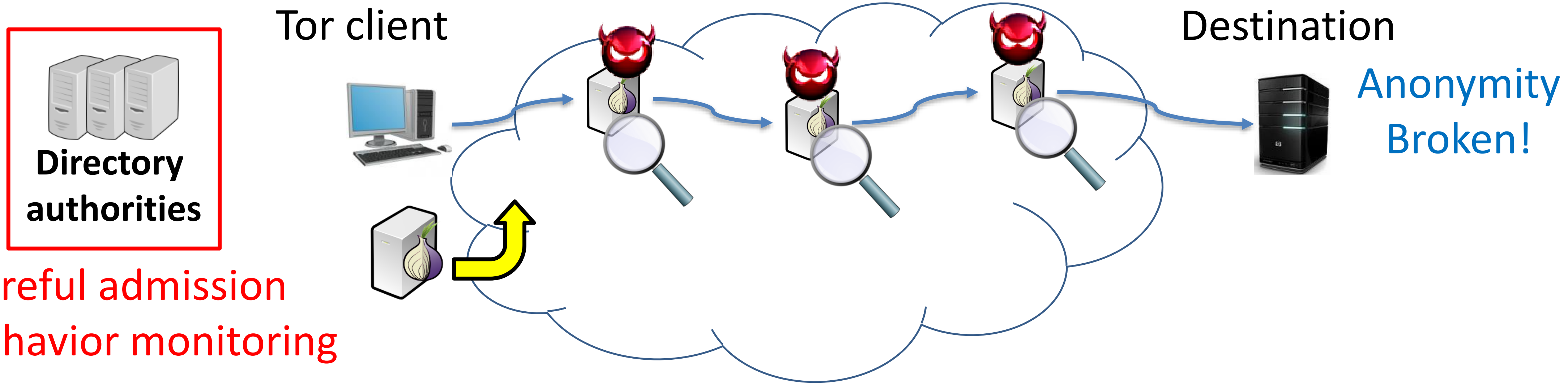
Can run a Tor relays of his own	Can compromise some fraction of Tor relays	Can observe some fraction of network traffic

Tor network: Threat model (Cont.)



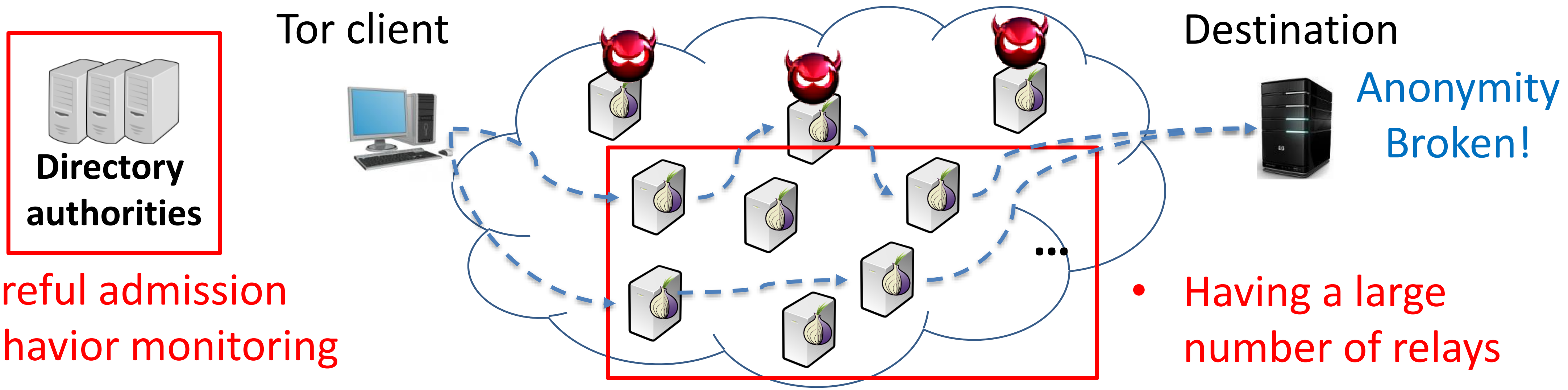
- Careful admission
- Behavior monitoring

Tor network: Threat model (Cont.)



- Careful admission
- Behavior monitoring

Tor network: Threat model (Cont.)



- Careful admission
- Behavior monitoring

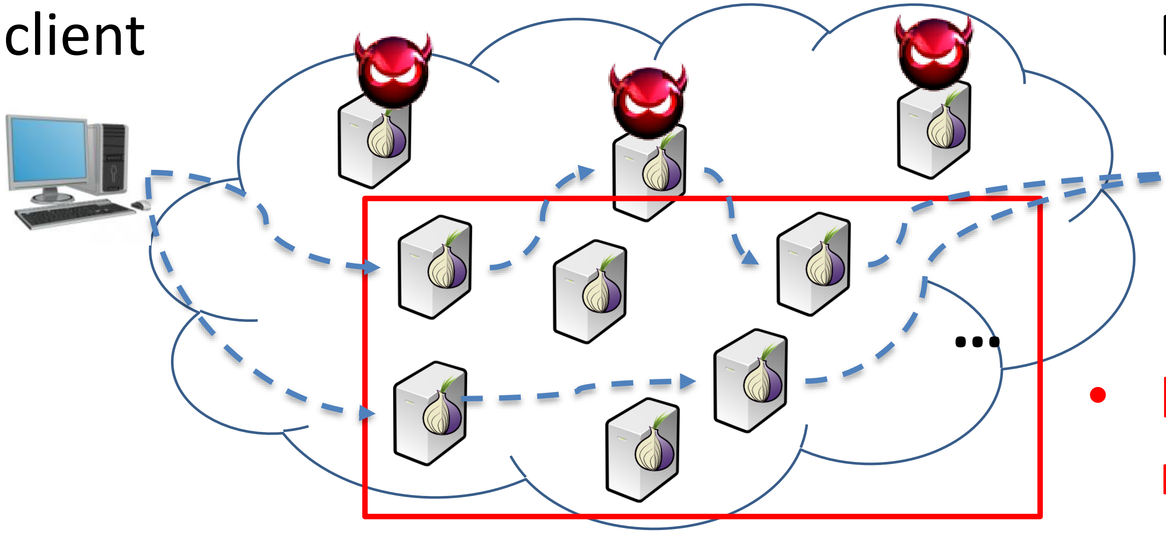
- Having a large number of relays

Out-of-scope: network-level adversary (controls a large fraction of network)

1. Currently runs ~10,000 relays
2. Large-scale traffic correlation is believed to be verify difficult in practice

Tor network: Threat model (Cont.)

- Careful admission
- behavior monitoring



- Having a large number of relays

However, Tor is still vulnerable to many types of attacks under its traditional threat model

Limitations of Tor

Problem 1. Tor relays are semi-trusted

– Authorities cannot fully verify the behaviors of them

Problem 2. Even attackers control a few Tor relays, they can

– Access internal information (circuit identifier, cell header, …)
– Modify the behavior of relays (DDoS, packet tampering, …)

<Low-resource attacks>

Modifying the behavior

- Malicious circuit creation [Security09, CCS11]
- Sniper attack [NDSS15]
- Bad apple attack [LEET11]

Both

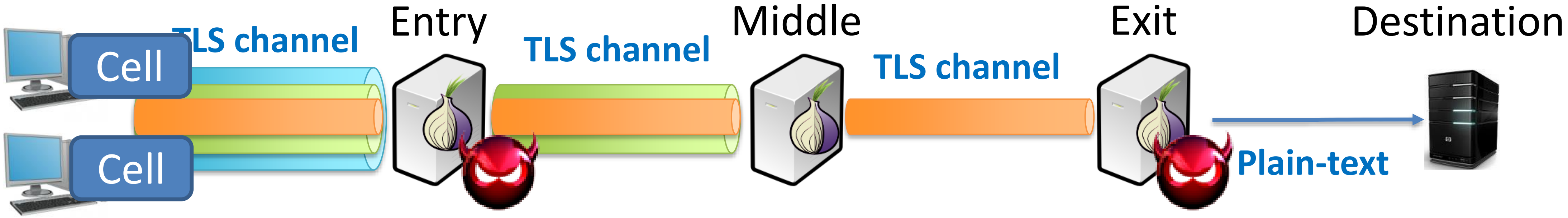
- tagging attack [ICC08, TON12, CCS12, S&P13]
- Bandwidth inflation [PETS07, S&P13]
- Controlling HSDir [S&P13]

Accessing internal information

- Harvesting hidden service descriptors [S&P13]
- Circuit demultiplexing [S&P06]
- Website fingerprinting [Security15]

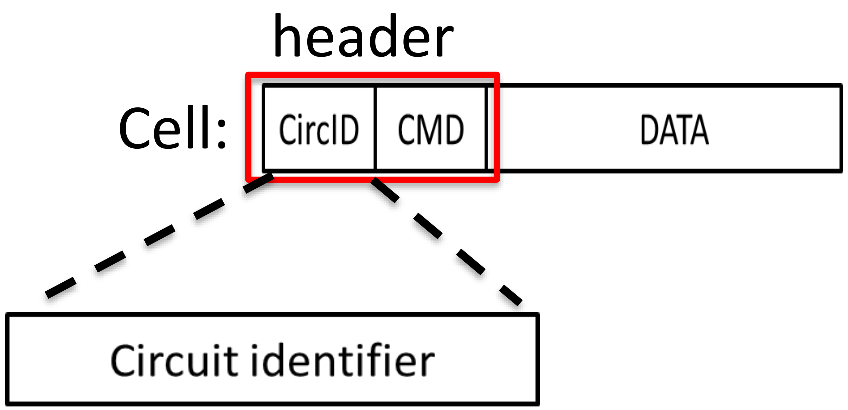
Limitations of Tor (Cont.)

Tor clients



Processing Unit: Cell (512 Bytes)

Information visible to attackers

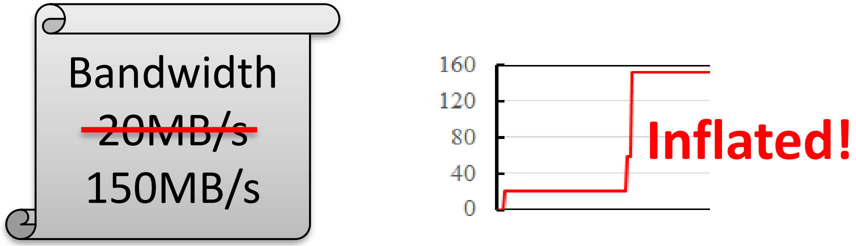


Demultiplex and identify a circuit

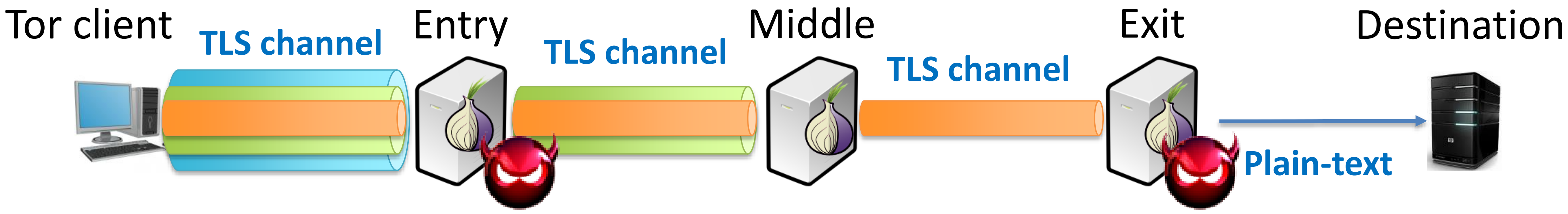
Attackers can modify the behavior

Modify or inject the cell

Give false information to others



Limitations of Tor (Cont.)



To address the problems on Tor,

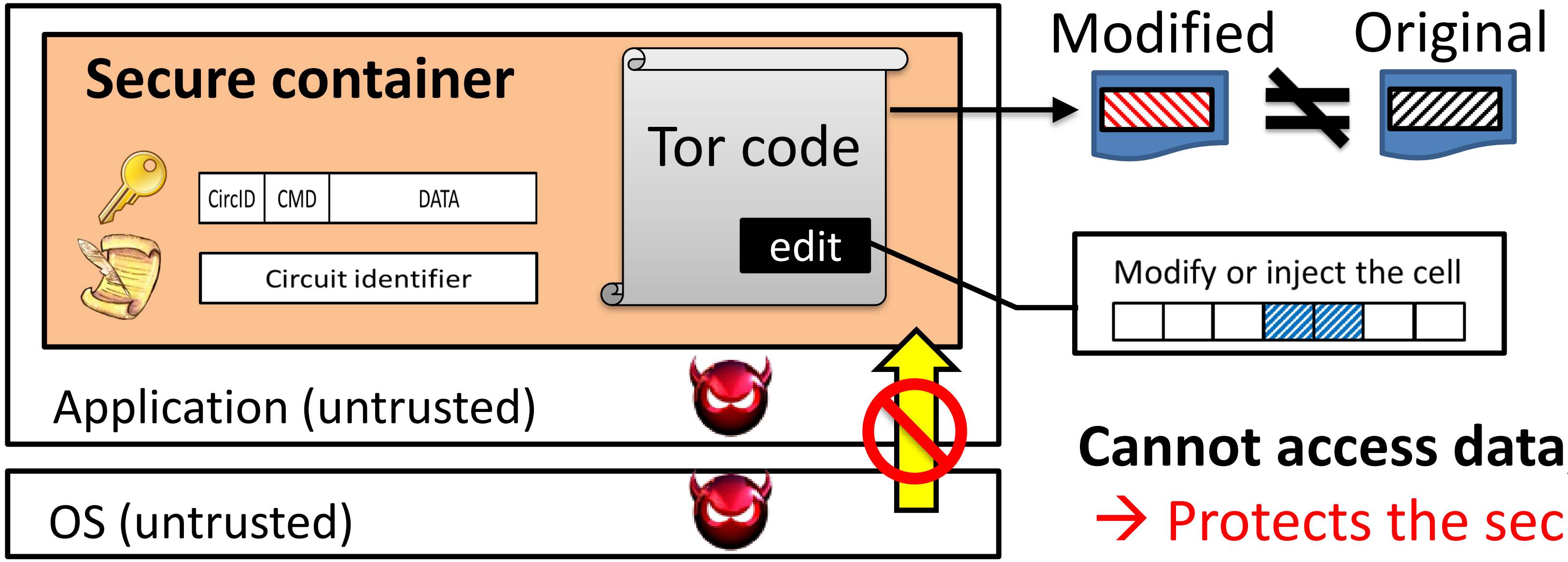
1) Fundamental trust bootstrapping mechanism

2) Advanced trust model to verify untrusted remote parties

are required

Trend: Commoditization of TEE

- Trusted Execution Environment (TEE): Hardware technology for trusted computing



Integrity checking
→ Prevents behavior modification

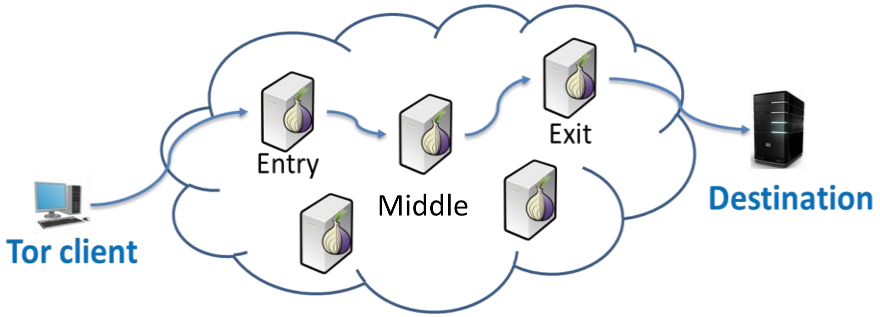
Cannot access data, flow control X
→ Protects the secrecy of the program

- **Intel SGX**: a promising TEE technology for generic applications
  - Native performance in the secure mode
  - Available on Intel Skylake and Kaby lake CPU

SGX-Tor: Leveraging Intel SGX on Tor

Intel SGX

+



Tor network

=

SGX-Tor

Improved trust model

Operational privacy

Practicality

Improved trust model

- Spells out what users trust in practice
- Provides ultimate privacy

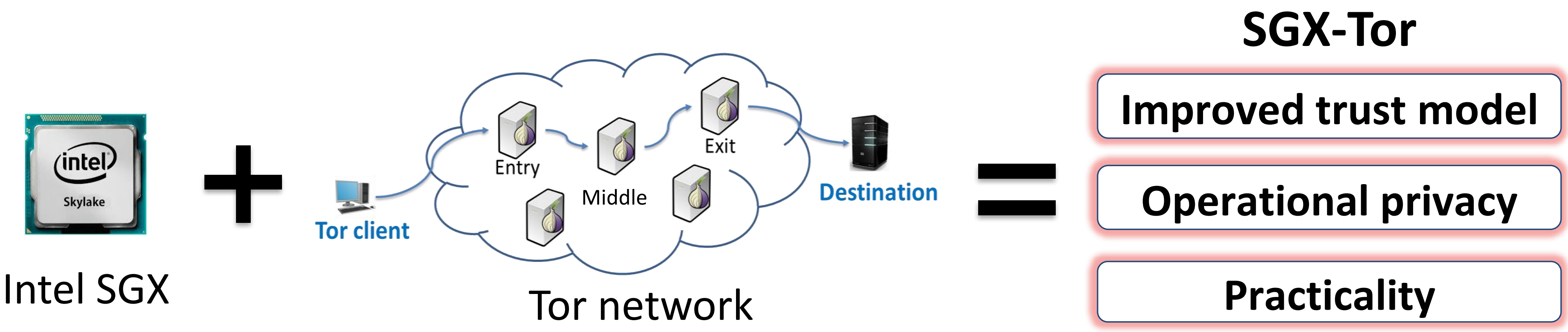
Operational privacy

- Protects sensitive data and Tor operations
- Prevents modifications on Tor relays

Practicality

- The chance of having more hardware resources donated
- Incrementally deployable
- Compatibility

SGX-Tor: Leveraging Intel SGX on Tor

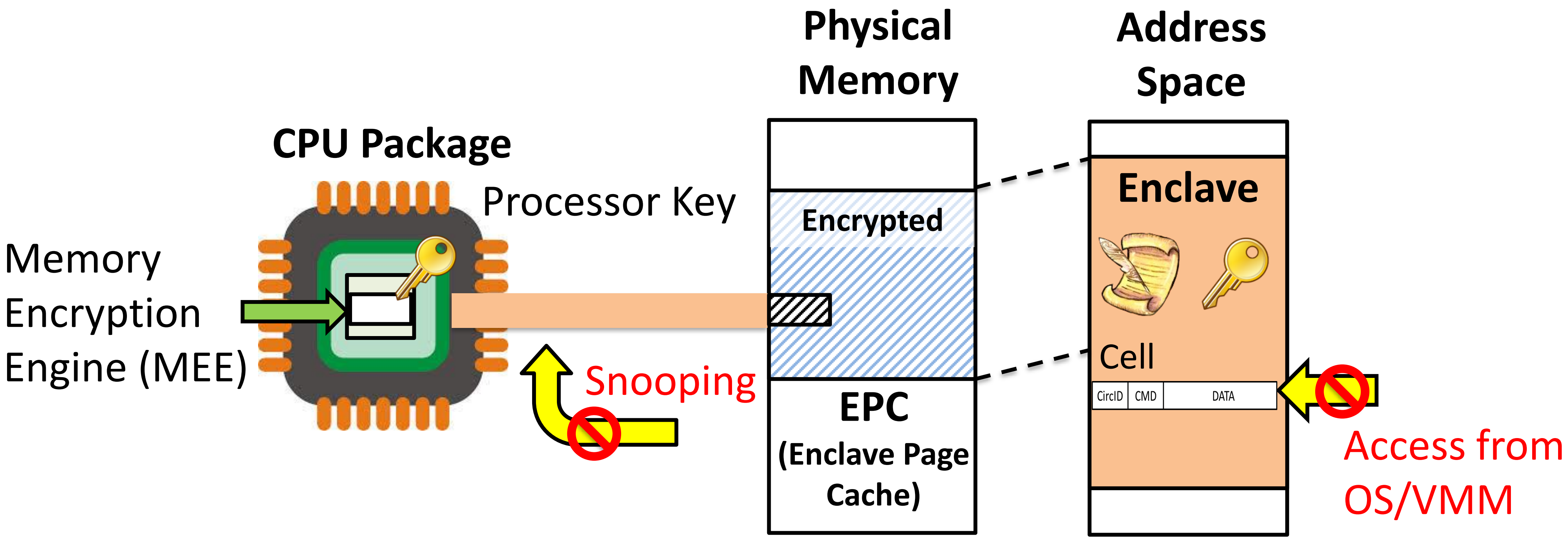


→ Reduces the power of an attacker who currently gets the sensitive information by running Tor relays

→ Raises the bar for Tor adversary to a traditional network-level adversary (only passively see the TLS bytestream)

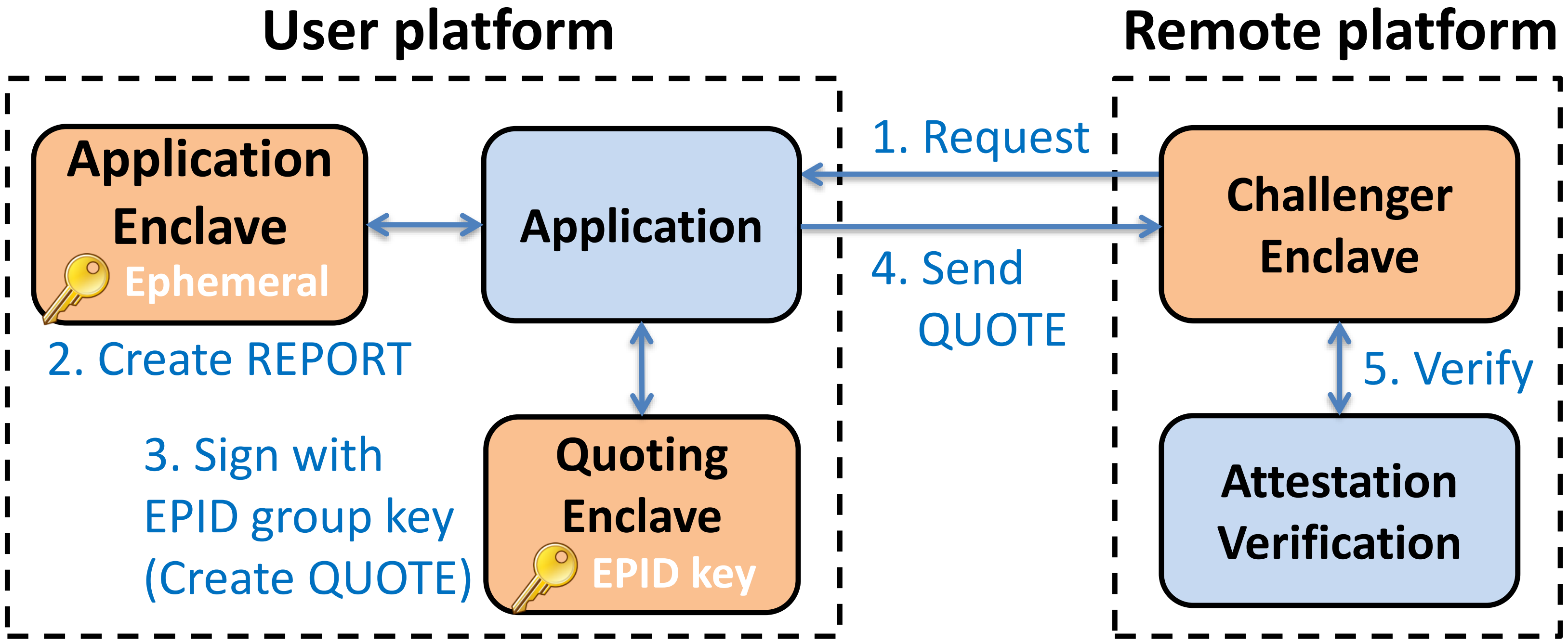
Intel SGX 101: Isolated Execution

- Protects app's secret from untrusted privilege software
- Application keeps its data/code inside the "**Enclave**"
- Trusted Computing Base (TCB) = Enclave + CPU package



Intel SGX 101: Remote attestation

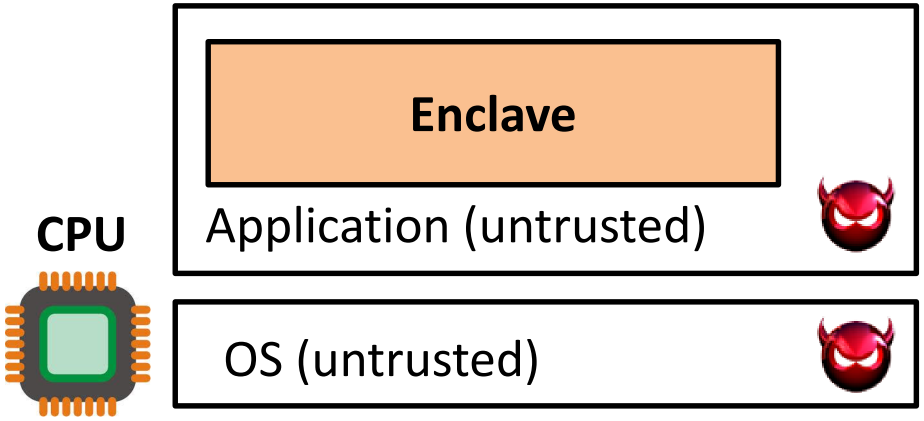
- Attest an application on remote platform
  - Checks the **integrity of enclave** (hash of code/data pages)
  - Verifies whether **enclave is running on real SGX CPU**
  - Can establish a "*secure channel*" between enclaves



20

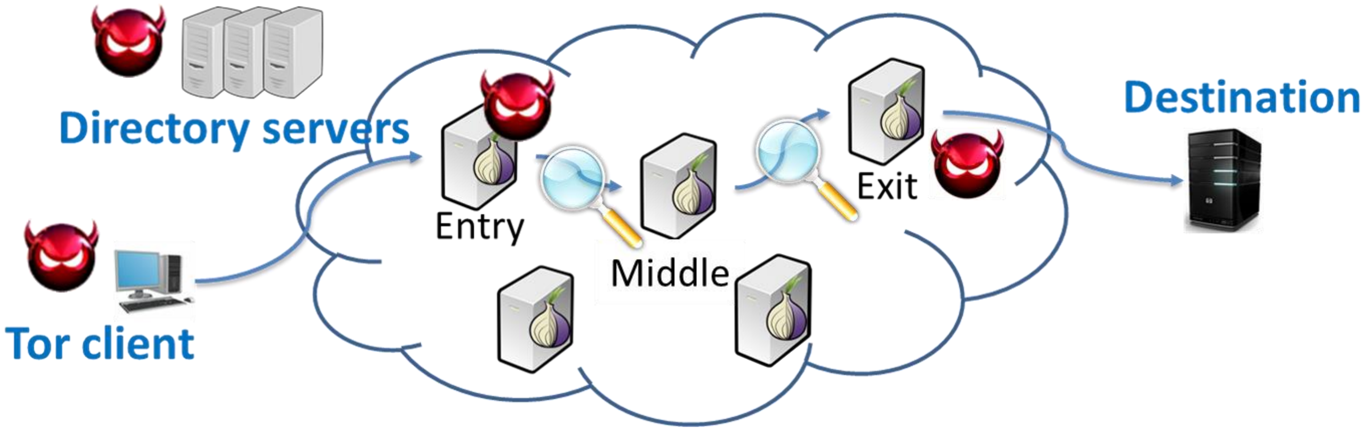
SGX-Tor: Threat Model

<SGX Threat model>



TCB : Enclave + CPU package

<Tor Threat model>



A powerful network-level adversary : out-of-scope

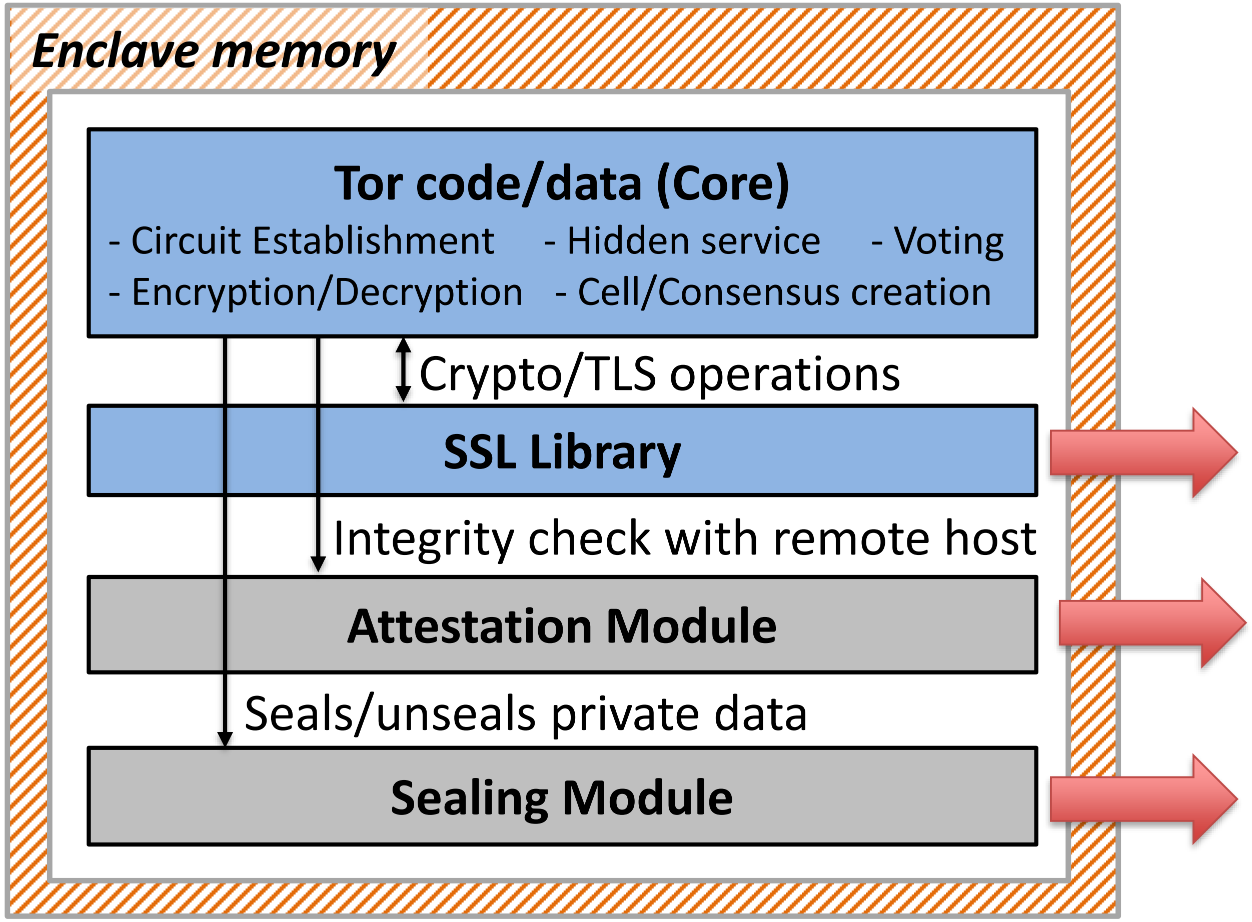
- Only trusts the underlying SGX hardware & Tor code itself
- Do not address network-level adversaries : who can perform large-scale traffic analysis

- Out of scope :Vulnerabilities in Tor codes, SGX side channel attacks

  → Mitigated by recent SGX research: Moat [CCS16], SGX-Shield [NDSS17], T-SGX [NDSS17]

SGX-Tor: Design and Implementation

User process (Tor application)

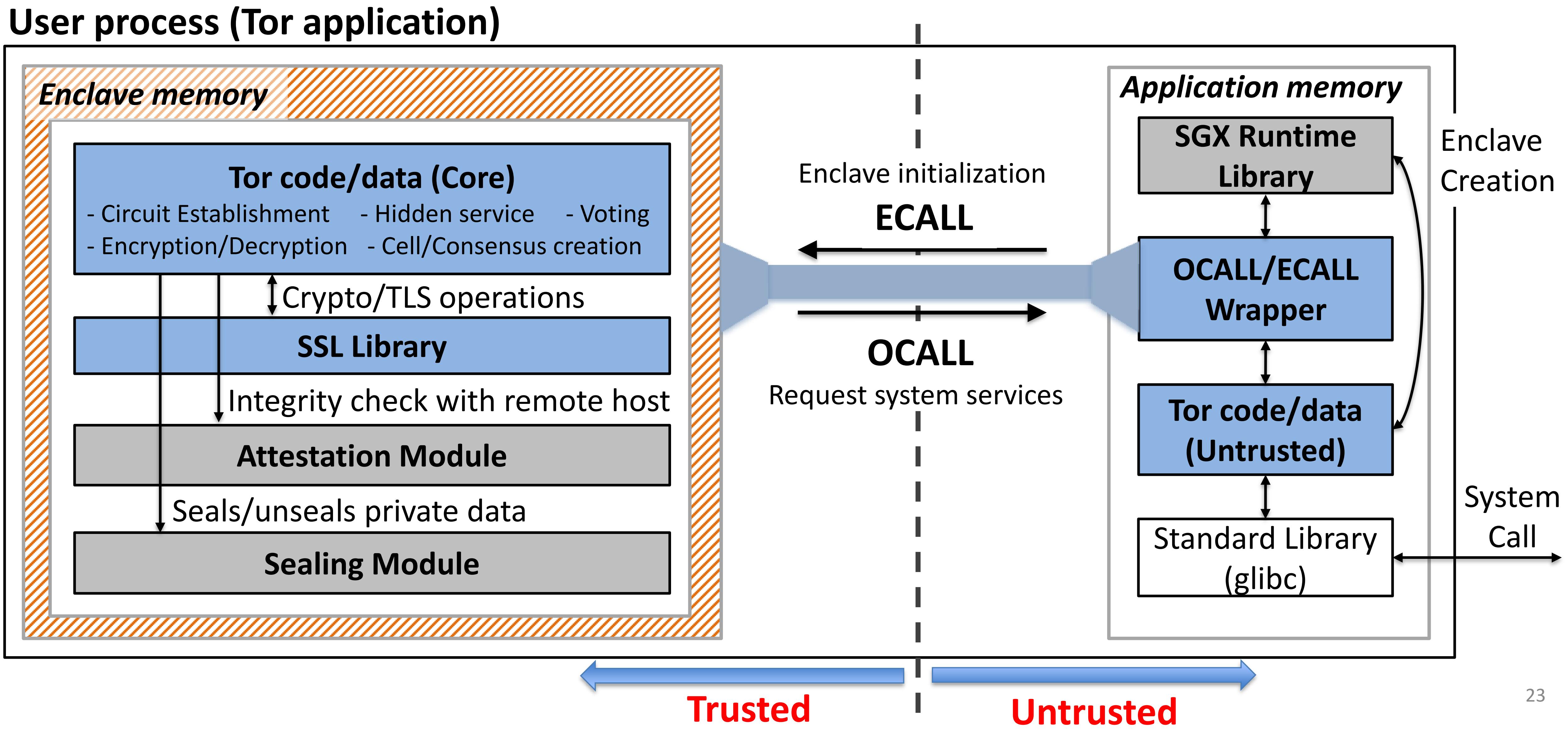


Securely obtains the entropy and time value

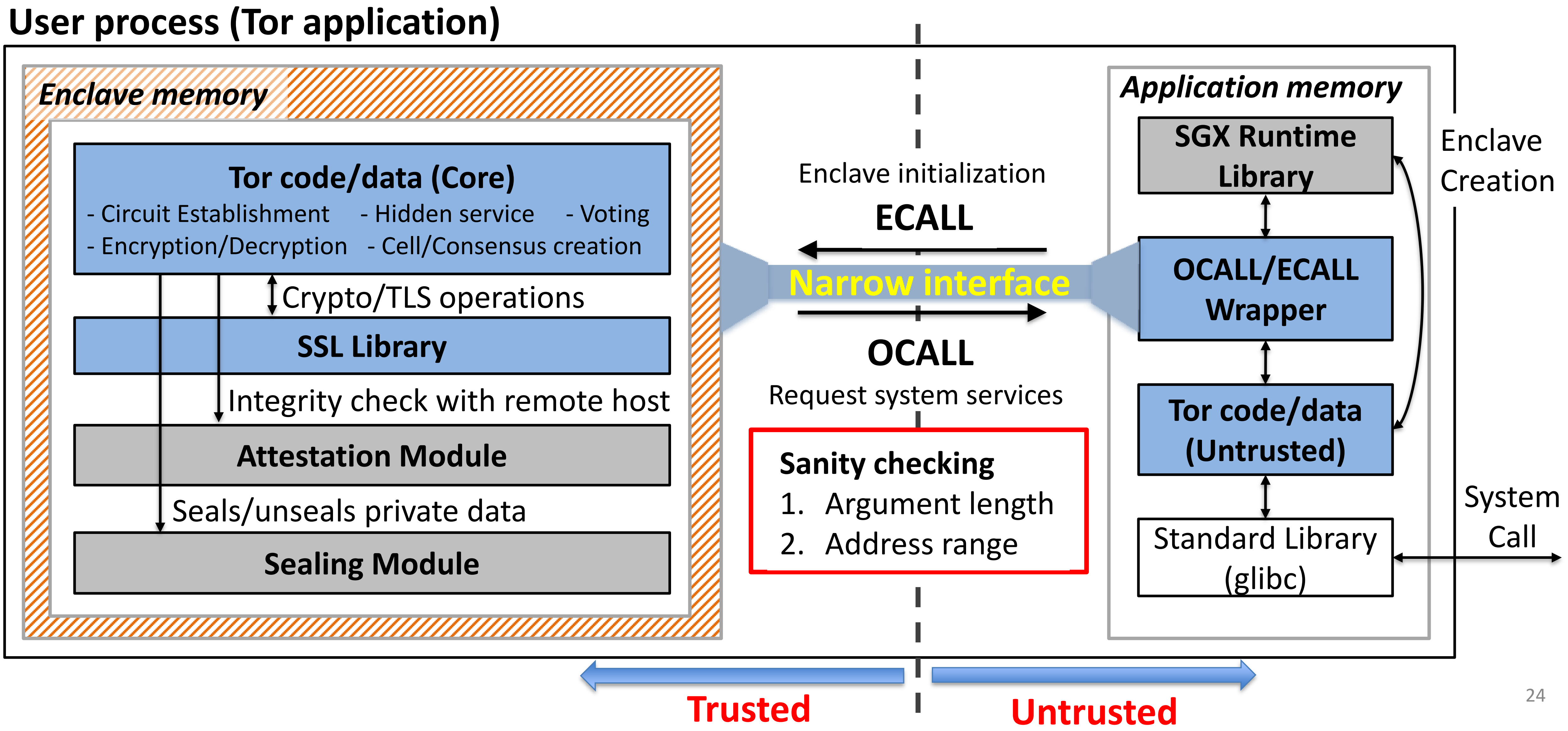
Validates the enclave hash of the Tor program

Encrypts and stores the sensitive data outside the enclave

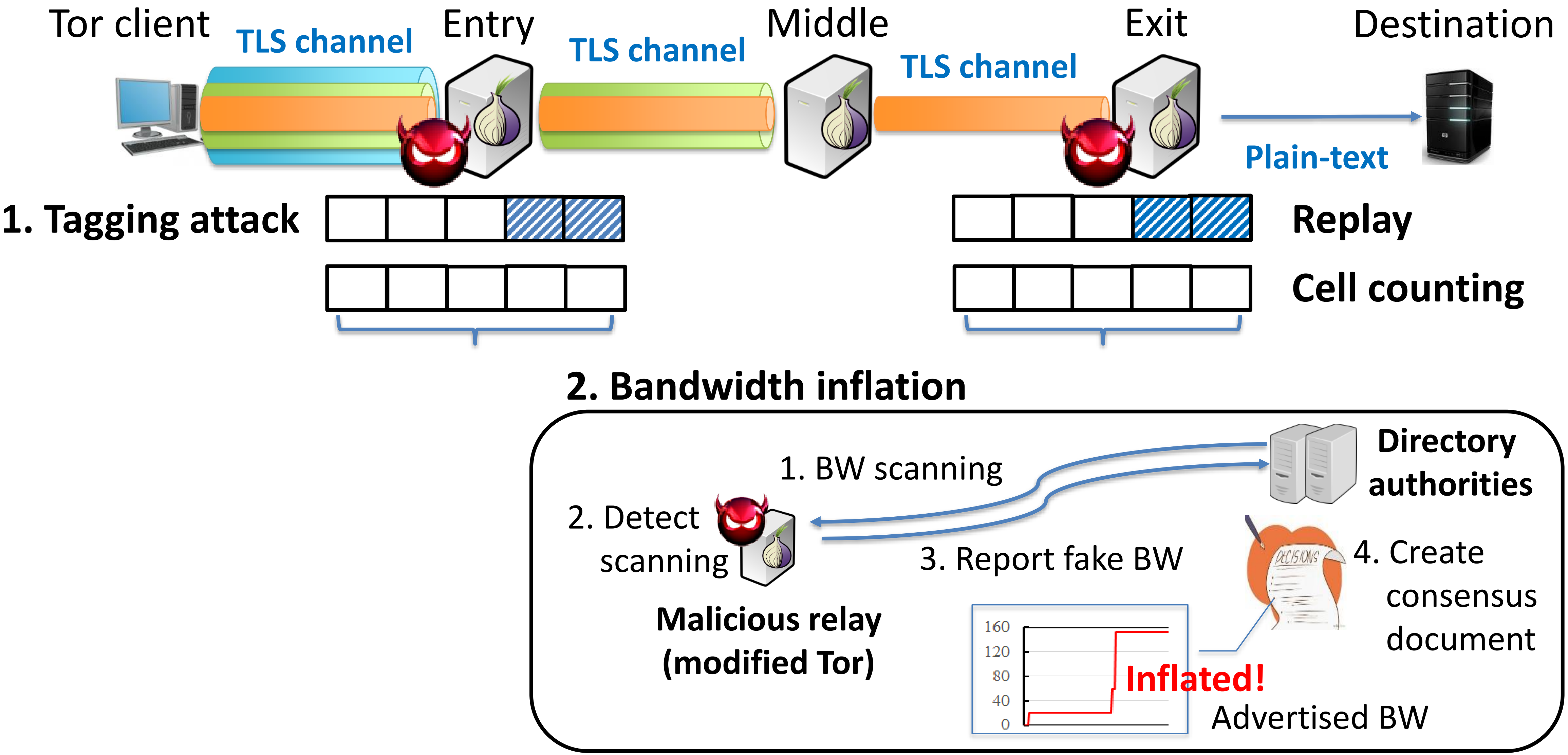
SGX-Tor: Design and Implementation



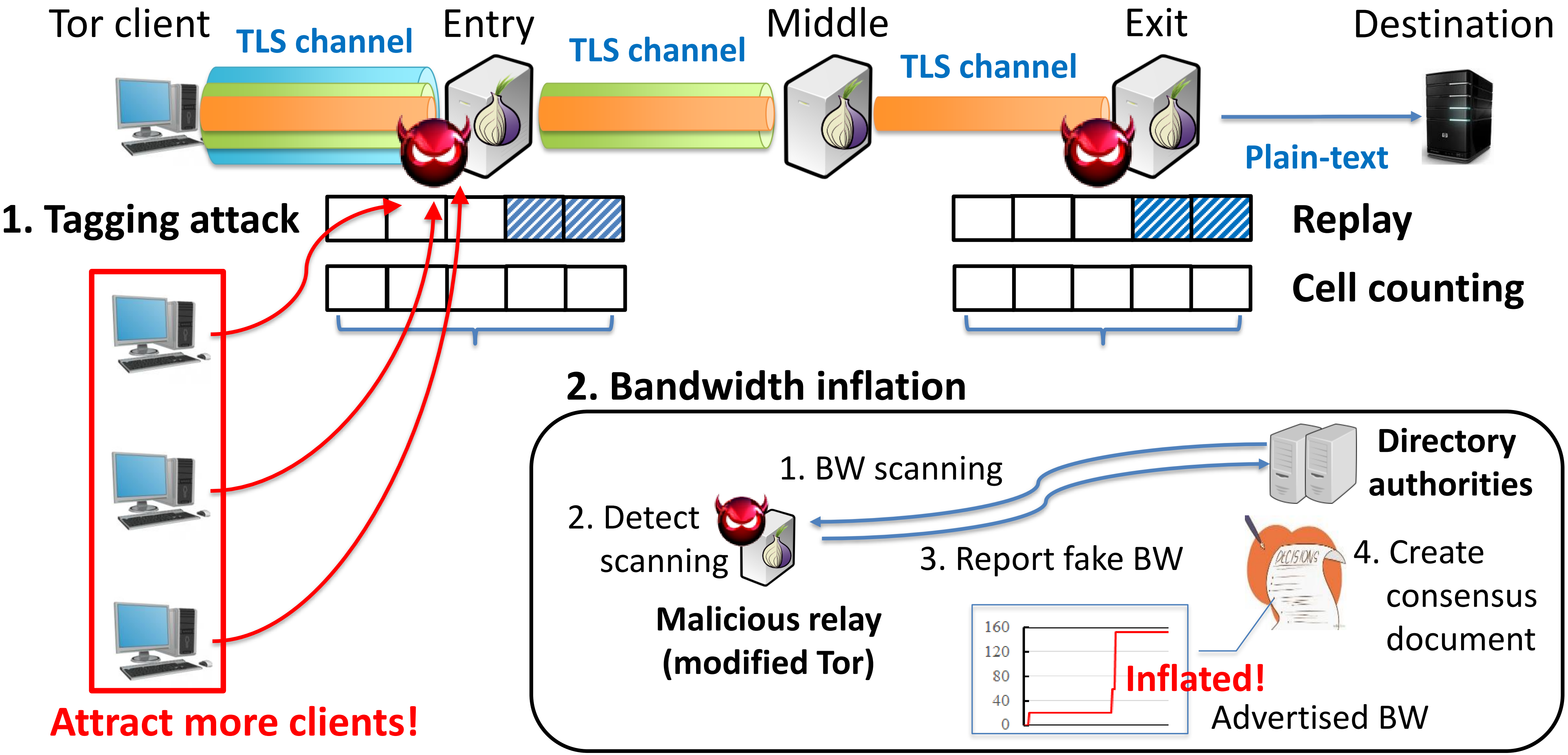
SGX-Tor: Design and Implementation



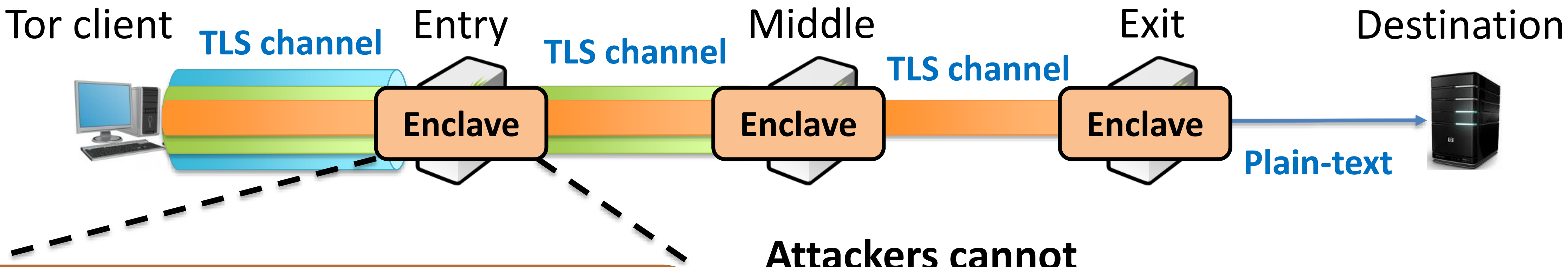
Attacks defeated by using SGX-Tor

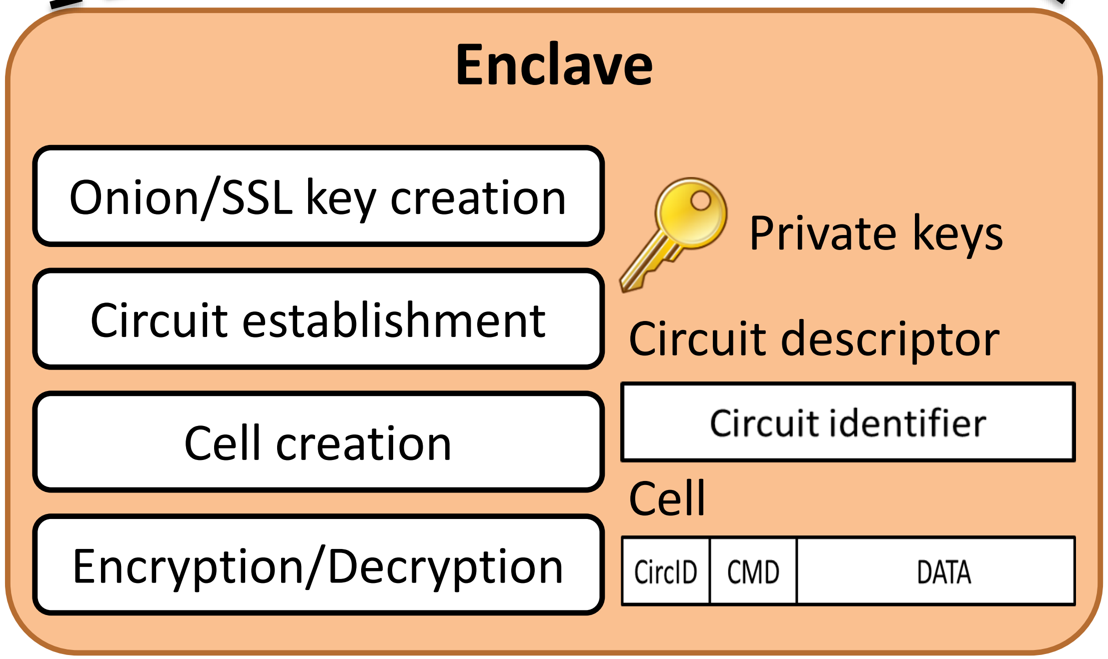


Attacks defeated by using SGX-Tor



Attacks defeated by using SGX-Tor (Cont.)





Attackers cannot

access
1. Circuit identifier
2. Cell header
3. Private keys

modify the code
1. To modify/inject cells
2. To inflate bandwidth

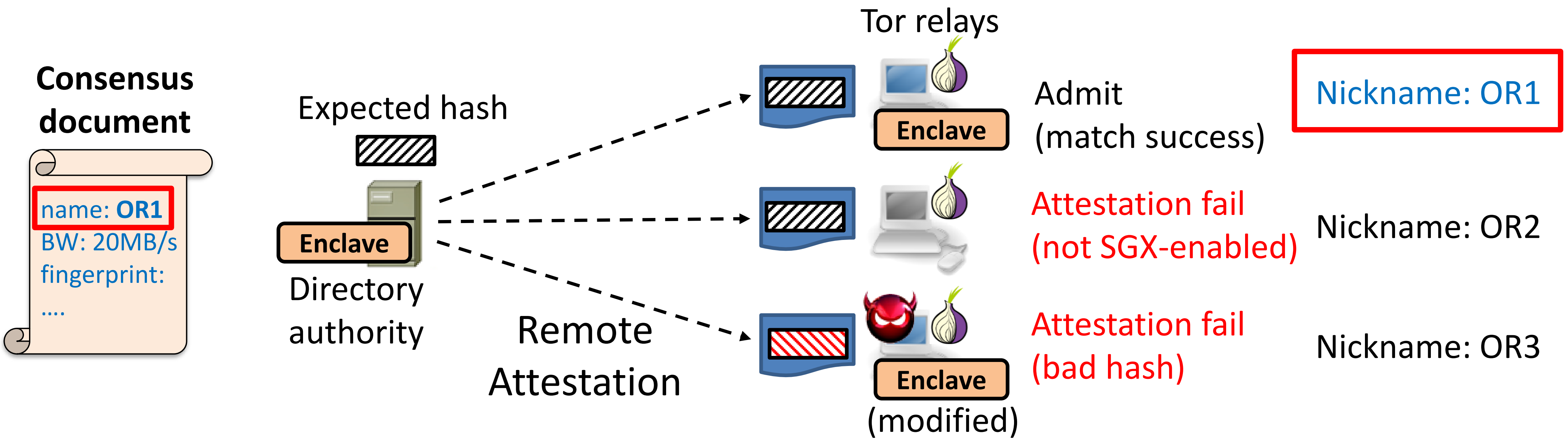
Attacks defeated/mitigated by SGX-Tor

- Circuit demultiplexing [S&P06]
- Bandwidth inflation [PETS07, S&P13]
- Harvesting/Controlling HSDir [S&P13]
- Tagging attack [ICC08, TON12, CCS12, S&P13]

…

New functionality: Automatic admission

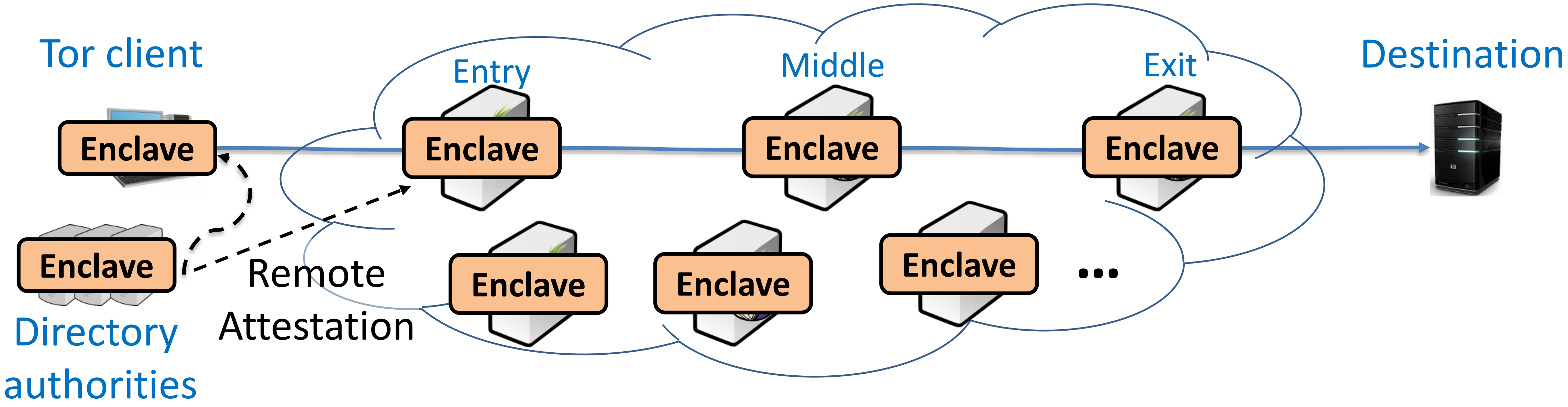
- **Integrity verification of relays (Directory authority → Onion Router)**
  - Automatically admits "**unmodified**" and "**SGX-enabled**" relays
  - **Improved trust model**: current implicit trust model turns into the explicit trust model



NOTE: Tor uses the same binary for directory authorities, Tor relays, and client proxies

Incremental deployability

- **SGX-Tor's basic assumption:** "All relays and authorities are SGX-enabled"
- **SGX-Tor supports interoperability**
  - Allows admission of non-SGX relays without remote attestation
  - SGX-enabled clients can get the list of SGX-Tor relays from SGX-enabled authorities



Implementation detail

- Engineering efforts
  - Support for Windows/Linux (based on Intel SGX SDK)
  - SGX-ported libraries: OpenSSL, libevent, zlibc
  - SGX-Tor is an open source: Available at https://github.com/KAIST-INA/SGX-Tor

- Trusted Computing Base (TCB) size
  - TCB size of Haven: More than 200MB (maximum enclave size : 128MB in Windows)
  - 3.8x smaller **(320K LoC vs 1,228K LoC)** than Graphene (open source library OS for SGX)

Evaluation

1) What kind of sensitive data of Tor is protected by SGX-Tor?
2) What is the performance overhead of running SGX-Tor?
3) How compatible and incrementally deployable is SGX-Tor with the current Tor network?

- Environmental setup
  - SGX CPUs: Intel Core i7-6700 (3.4GHz) and Intel Xeon CPU E3-1240 (3.5GHz)
  - Configuration: 128MB Enclave Page Cache (EPC)
  - Running Tor in Windows, Firefox as a Tor browser (in the client proxy)
  - Establish a private Tor network using *chutney*

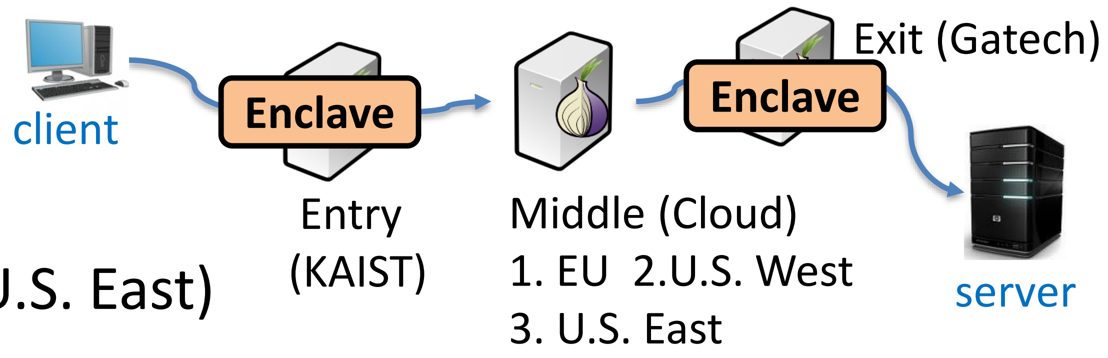
What is protected by SGX-Tor?

	Current Tor	Network-level adversary	SGX-Tor
TCP/IP header	Visible	**Visible**	**Visible**
TLS-encrypted bytestream	Visible	**Visible**	**Visible**
Cell	Visible	Not visible	Not visible
Circuit ID	Visible	Not visible	Not visible
Voting result	Visible	Not visible	Not visible
Consensus document	Visible	Not visible	Not visible
Hidden service descriptor	Visible	Not visible	Not visible
List of relays	Visible	Not visible	Not visible
Private keys	Visible	Not visible	Not visible

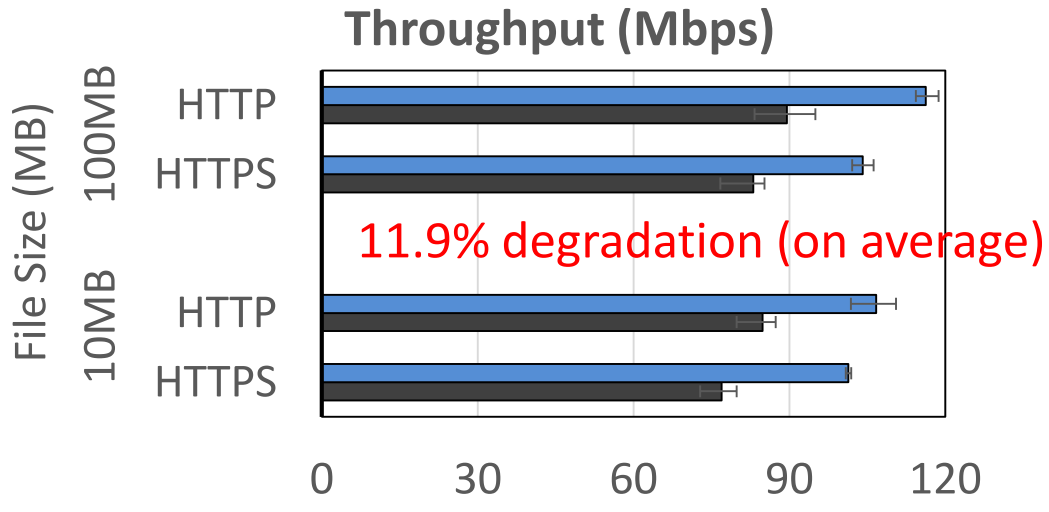
Performance evaluation

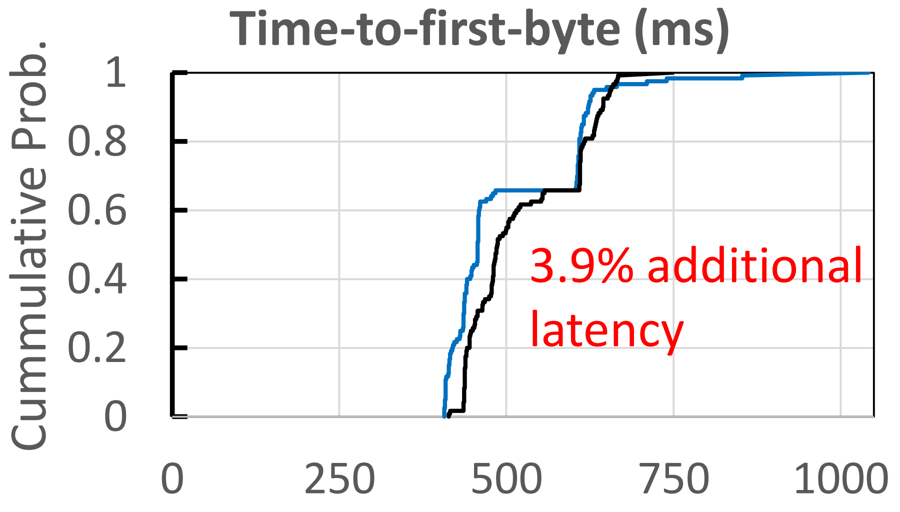
- SGX-Tor performance : WAN setting
  - Establish a private Tor network
  - For the realistic scenario, we consider the "**locality of relays**" (Asia, EU, U.S. West, U.S. East)

<Evaluation environment>



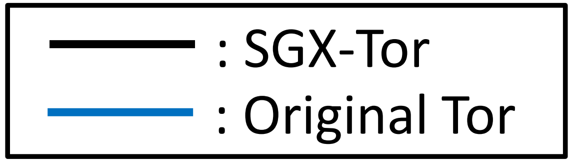
: SGX-Tor : Original Tor

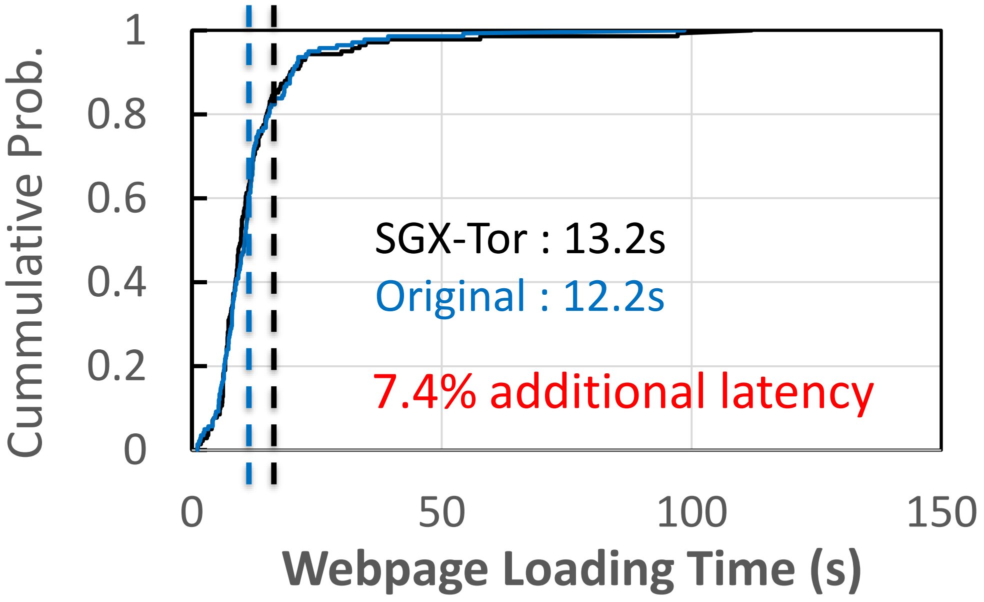


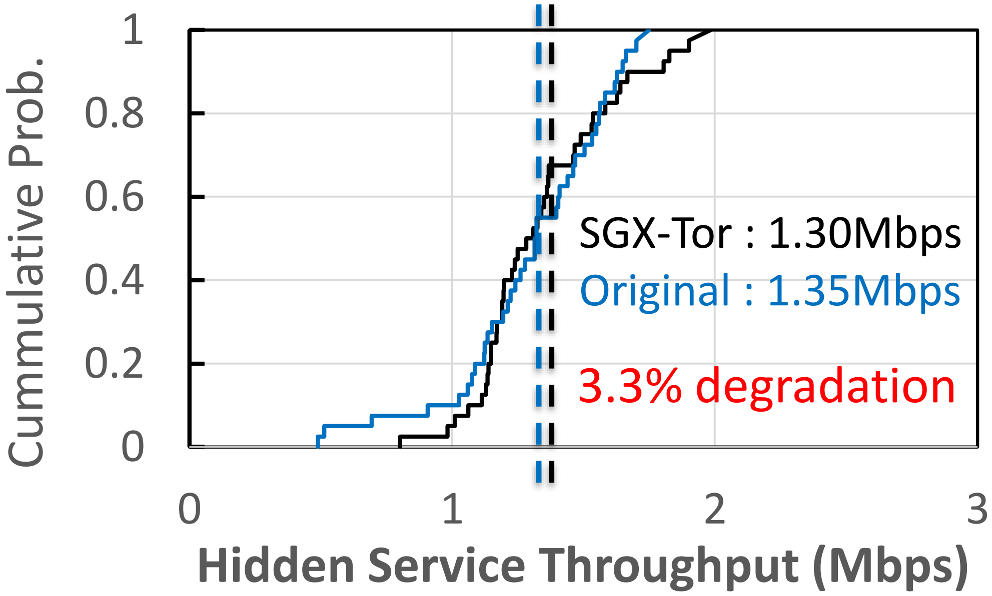


Performance evaluation (Cont.)

- End-to-end client performance of SGX-Tor (using Tor browser)
  - Web latency: Visiting Alexa Top 50 websites
  - Hidden service: HTTP file server (downloading 10MB)

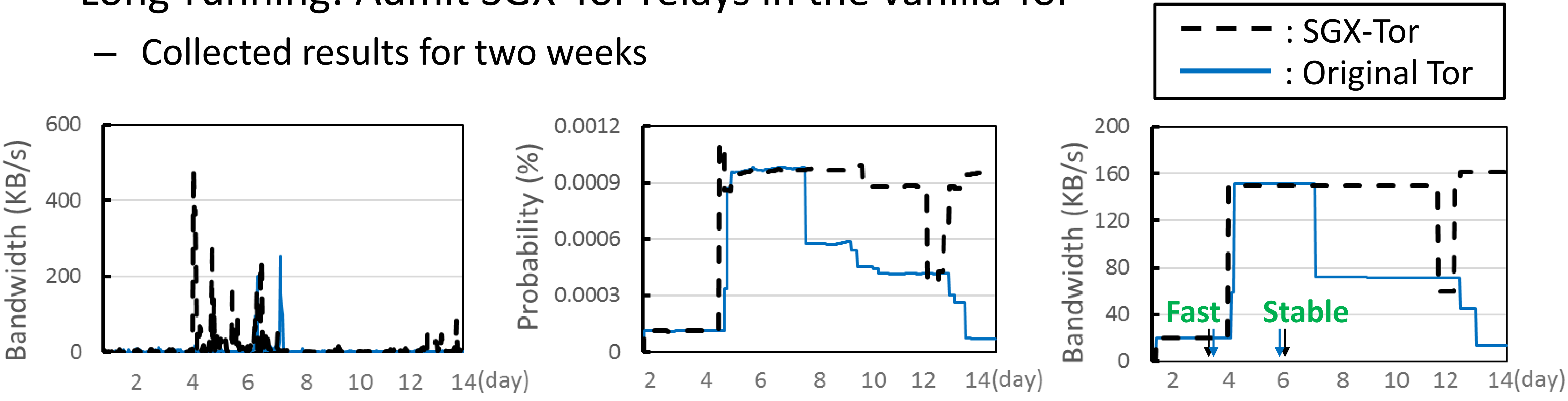






Compatibility with vanilla Tor

- Long-running: Admit SGX-Tor relays in the vanilla Tor
  - Collected results for two weeks



Network I/O bandwidth per second — Serves Tor traffic well

Middle selection Probability — Actually selected by multiple Tor users

Advertised bandwidth * — Listed in the consensus document

* From https://collector.torproject.org/

Conclusion

- We design and implement SGX-Tor by leveraging commodity TEE and demonstrate its viability
  - Gives moderate performance overhead
  - Shows its compatibility and possibility of incremental deployment
- SGX-Tor enhances the security and privacy of Tor by
  - Defending against existing attacks on Tor
  - Bringing changes to the trust model of Tor
  - Providing new properties : automatic admission
- Available at github! (https://github.com/KAIST-INA/SGX-Tor)