A First Step Towards Leveraging Commodity Trusted Execution Environments for Network Applications

Seongmin Kim  Youjung Shin  Jaehyung Ha
Taesoo Kim*  Dongsu Han

KAIST  * Georgia Tech

HotNets 2015, Dongsu Han
Trend 1: Security and Privacy
Critical Factors in Technology Adoption

• Demands for “security” and “privacy” are increasing
  – Widespread use of Transport Layer Security (TLS)
  – Popularity of anonymity networks (e.g., Tor)
  – Use of strong authentication/encryption in WiFi

• Expectation on security and privacy impacts design decisions:
  – Operating system (iOS, Android)
  – Apps/services (e.g., messenger, adblocker)
  – Network infrastructure (inter-domain SDN)
Demands for "security" and "privacy" are increasing – Widespread use of Transport Layer Security (TLS) – Popularity of anonymity networks (e.g., Tor) – Use of strong authentication/encryption in WiFi

Expectations on security and privacy impacts design decisions: – Operating system (iOS, Android) – Apps/services (e.g., messenger, adblocker) – Network infrastructure (inter-domain SDN)

Trend 1: Security and Privacy Critical Factors in Technology Adoption
Trend 2: Commoditization of Trusted Execution Environment

• Trusted Execution Environment (TEE)
  – Isolated execution: integrity of code, confidentiality
  – Remote attestation

• Commoditization of TEE

The commoditization of TEE brings new opportunities for network applications.

2. Compatibility with x86
Network Applications + TEE = ?

- What impact does TEE have on networking?

- Previous efforts: Adopting TEE to cloud platform
  - Haven [OSDI’14]: Protects applications from an untrusted cloud
  - VC3 [S&P’15]: Trustworthy data analytics in the cloud
SGX: Isolated Execution

- Application keeps its data/code inside the “enclave”
  - Smallest attack surface by reducing TCB (App + processor)
  - Protect app’s secret from untrusted privilege software (e.g., OS, VMM)
SGX : Remote Attestation

- Attest an application on remote platform
- Check the identity of enclave (hash of code/data pages)
- Can establish a “secure channel” between enclaves

1. Request
2. Calculate MAC
3. Send MAC
4. Verify
5. Sign with group key [EPID]
6. Send signature

HotNets 2015, Dongsu Han
Case Studies: Three Applications

1. **Network infrastructure**: Software-defined inter-domain routing
2. **Peer-to-peer systems**: Tor anonymity network
3. **Middlebox**: TLS and “secure” middleboxes
SDN-based Inter-domain Routing

1) Topology
2) Policies (e.g., export policy)

• Offers new properties
  – Fast convergence, application-specific peering, flexibility, what-if analysis [hotnets2011]

• Reveals private information: topology and policy
Prior work [hotnets2011] uses Secure Multi-Party Computation (SMPC) to solve this, but the computational complexity is prohibitive.
Enclose private information inside the enclave

Communication through a secure channel after attestation
ASes agree upon a common code base. Makes sure that it does not leak private information [Moat]. It becomes the TCB of the inter-domain routing infrastructure.
SDN-based Inter-domain Routing

1. Mutually attest/authenticate using remote attestation
2. Collect policy and topology through a secure channel
3. Main controller computes routing path
4. Sends routes for each AS through a secure channel
Extending Features: Policy verification

- Enabling verification on routing decisions
  - Want to verify whether the promise is being kept [SPIDeR]
Extending Features: Policy verification

Among all routes to Google from B, is the one B advertising to A the shortest?

I will give you my shortest route to Google! [SPIDeR]

• Enabling verification on routing decisions
  – Want to verify whether the promise is being kept [SPIDeR]

• Verify only the predicates agreed upon by A and B

Inter-domain Controller

Enclave

Predicates

AS A

AS B

AS C

AS D
Tor: Anonymity Network

- Tor network: uses 3-hop onion routing
  - Directory servers: Advertise available onion routers (ORs), vote for bad exit nodes
  - Relies on volunteer provided hosts

When exit node is compromised, (unless end-to-end encryption is used)
1. Snooping or tampering of the plain-text
2. Break of anonymity: Bad apple attack

Tor network

Directory servers

Tor client

HotNets 2015, Dongsu Han
Tor: Anonymity Network

- Tor network: uses 3-hop onion routing
  - Directory servers: Advertise available onion routers, vote for bad exit nodes
  - Relies on volunteer provided hosts

When directory servers are compromised,
1. Tie-breaking attacks while voting
2. Admission of malicious ORs

Directory servers
Tor client
Tor network
Entry
Relay
Exit
Destination
HotNets 2015, Dongsu Han
Application of TEE to Tor

1) SGX-enabled directory servers
2) SGX-enabled directory servers & ORs

- Keep sensitive data in enclave
  1) Authority keys
  2) List of available ORs
- Integrity check each other
- Automatic admission of new ORs

- Detect problematic exit nodes through integrity check
Application of TEE to Tor

1) SGX-enabled directory servers
2) SGX-enabled directory servers & ORs
3) Fully SGX-enabled setting
   → Eliminate directory servers altogether

Each Tor components can check the integrity of target program (Tor binary)
Implementation

- OpenSGX [NDSS’16] : Open source SGX emulator
  - Fully functional, instruction-compatible emulator of SGX build on top of QEMU
  - Emulates system software and provide SGX libraries
Preliminary Evaluation: Overhead

• Estimate the overhead in terms of additional CPU cycles
  – Each SGX instruction = 10 k cycles [Haven]

<table>
<thead>
<tr>
<th>Target</th>
<th>QuoRng</th>
<th>Challenger</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Cycles (B)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3%</td>
<td>0</td>
</tr>
</tbody>
</table>

Cost of remote attestation:
3% of 1024-bit Diffie-Hellman
SDN-based inter-domain routing

- 30 ASes with the centralized inter-domain controller
- Inter-domain controller: 90% more CPU cycles
- AS-local controllers: 70% more CPU cycles

<# of CPU cycles consumed in the inter-domain controller>

Number of participating ASes: 30

- With SGX
- Without SGX

CPU Cycles (M)

HotNets 2015, Dongsu Han
Conclusion

• Commoditization of TEE brings new opportunities for network applications

• Cases studies show wide range of impact:
  – Policy privacy of SDN-based inter-domain routing
  – New design space of Tor anonymity network
  – Secure in-network functions

• SDN-based inter-domain routing:
  – Characterize and measure the overhead of using SGX
  – Consumes $70-90\%$ more CPU cycles
SDN-based inter-domain routing

- 30 ASes with the centralized inter-domain controller
- Inter-domain controller: 90% more CPU cycles
- AS-local controllers: 70% more CPU cycles

<# of CPU cycles consumed in the inter-domain controller>

![Graph showing CPU cycles vs. number of participating ASes]
Secure Multi-party Execution

- SGX Program owner can remotely verify the integrity of code
- Publicly available programs (e.g., git) can validate the integrity of project by sharing the private key for the attestation
- Creates signature of program through shared private key

Example:
1. Tor binary for OR
2. SDN-based controller program
In-network Functions (Middleboxes)

- Use of TLS protocol disrupts in-network processing
  - Only endpoints of communication can access the plain-text
- SGX enables opportunity for secure in-network functions

Can be done unilaterally or bilaterally.
Trend 2: Commoditization of Trusted Execution Environment

- Trusted Execution Environment (TEE)
  - Isolated execution: integrity of code, confidentiality
  - Remote attestation

- Commoditization of TEE

The commoditization of TEE brings new opportunities for network applications.