Performance evaluation in trust enhanced decentralised content distribution networks

Roman Korn¹, Nicolai Kuntze¹, Juergen Repp¹

¹Fraunhofer Institute for Secure Information Technology (SIT)
Darmstadt, Germany

May 10, 2011
Outline

1. Background
2. Challenge
3. Approach and Findings
The current model: Data centers
The nano data center model
The nano data center model

The Internet

Video server

Control server

DSLAM

RHGs

Roman Korn, Nicolai Kuntze, Juergen Repp

Performance evaluation in trust enhanced decentralised content distribution networks
Challenges from the IT security perspective

- Devices are deployed at the households of end customers
- Content is not bound to specific boxes (VoD use case) or not even existing beforehand (online gaming use case)
- Centralised threat mitigation is slow and costly
Trust establishment

- Aim is the establishment of a self-healing network under the peril of malicious nodes
- Introduction of an hardware module to establish a **Root of Trust** to establish trustworthiness
- Application of standardised industry approach by using the **Trusted Platform Module**
- Design of specialised protocols addressing the characteristics of the TPM and its protocols
Extended Tracker Protocol

0. Setup previous to protocol.

\[ p : \text{metafile}, \left( \text{AIK}^p_{\text{pub}}, \text{AIK}^p_{\text{priv}} \right), \text{AIKCert}^p_{\text{ca}}, S^p_{\text{pub}} \]

\[ t : \text{metafile}, \left( S^t_{\text{pub}}, S^t_{\text{priv}} \right), C^t_{\text{pub}} \]

1. Peer sends an encrypted request.

\[ p \rightarrow t : \text{enc} \left\{ \text{request}, K^p_{\text{pub}}, \text{SML}^p, \text{AIKCert}^p_{\text{ca}} \right\} \]

2. Tracker computes the shared secret.

\[ t : K^p_{\text{pub}} \circ K^t_{\text{pub}} \]

3. Tracker sends a signature and a public key.

\[ t \rightarrow p : \text{sig} \left\{ \text{hash} \left( K^p_{\text{pub}} || K^t_{\text{pub}} \right) \right\}_{S^t_{\text{pub}}}, K^t_{\text{pub}} \]

4. Peer sends a quote.

\[ p \rightarrow t : \text{quote} \left\{ \text{hash} \left( K^t_{\text{pub}} || K^p_{\text{pub}} \right), \text{PCR}_n \right\}_{\text{AIK}_n^p} \]

5. Peer computes the shared secret.

\[ p : K^p_{\text{pub}} \circ K^t_{\text{pub}} \]

6. Tracker sends the encrypted response.

\[ t \rightarrow p : \text{enc} \left\{ \text{Data} \right\}_{K^t_{\text{pub}}} \]

Extended Peer-Wire Protocol

0. Setup previous to protocol.

\[ pA : \left( \text{AIK}^{pA}_{\text{pub}}, \text{AIK}^{pA}_{\text{priv}} \right), \text{data} := \left( \text{Address}_{\text{pub}}, \text{AIKCert}^{pA}_{\text{ca}} \right) \]

\[ pB : \left( \text{AIK}^{pB}_{\text{pub}}, \text{AIKCert}^{pB}_{\text{ca}} \right), K^{tA,1}_{\text{pub}}, \text{info_hash} \]

1. Peer pA sends the initial request (handshake).

\[ pA \rightarrow pB : \text{peerID}^{pA}, K^{tA}_{\text{pub}} \circ \text{ticket} := \text{enc} \left\{ \text{AIKCert}^{pA}_{\text{ca}}, \text{resource, time} \right\}_{K^{tA,1}_{\text{pub}}} \]

2. Peer pB sends a response.

\[ pB \rightarrow pA : K^{tA}_{\text{pub}}, \text{quote} \left\{ \text{hash} \left( K^{pA}_{\text{pub}} || K^{pA}_{\text{pub}} \right), \text{PCR}_0 \right\}_{\text{AIK}_{pA}^{tA}} \]

3. Peer pA sends a request.

\[ pA \rightarrow pB : \text{quote} \left\{ \text{hash} \left( K^{tA}_{\text{pub}} || K^{pA}_{\text{pub}} \right), \text{PCR}_0 \right\}_{\text{AIK}_{pA}^{tA}} \]

Peer pA computes the shared secret.

\[ pA : K^{tA,2}_{\text{pub}} \circ K^{pA}_{\text{pub}} \]

5. Peer pB computes the shared secret.

\[ pB : K^{tA,2}_{\text{pub}} \circ K^{pA}_{\text{pub}} \]

6. Peer pB sends the final encrypted response (handshake).

\[ pB \rightarrow pA : \text{enc} \left\{ \text{Content} \right\}_{K^{tA,2}_{\text{pub}}} \]
Tracker Measurements

(a) tBittorrent Tracker-Protocol

(b) jBittorrent Tracker-Protocol
Peer Wire Measurements

(a) tBittorrent Peer-Wire-Protocol

4,205 ms

0,977 ms

1,092 ms

0,080 ms

\{ peer pA peer pB \}

\{ updated \}

(1)

(2)

(3)

(4)

(5)

\{ \}

\{ \}

0,041 ms

0,009 ms

hs(resID, peerID)

hs(resID, peerID)

pA (4) = 0,032 ms

pB (4) = 0,024 ms

(b) jBittorrent Peer-Wire-Protocol

0,041 ms

0,009 ms

hs(resID, peerID)

hs(resID, peerID)
Measurements and distribution
BitTorrent performance

- **Background**
- **Challenge**
- **Approach and Findings**

Roman Korn, Nicolai Kuntze, Juergen Repp

Performance evaluation in trust enhanced decentralised content distribution networks
Conclusion

- Our trusted BitTorrent shows that security means can be incorporated.
- Hardware based security allows for certain optimisations in the integration of security functionalities.
- Security integration requires insight in the deployment environment.