Security vs. Performance: Tradeoffs using a Trust Framework

Aameek Singh
Kaladhar Voruganti
Sandeep Gopisetty
David Pease
Linda Duyanovich
Ling Liu

College of Computing
Georgia Tech
aameek@cc.gatech.edu

Storage Systems
IBM Almaden Research Center
kaladhar@us.ibm.com

Security and Performance

- Mostly contradictory goals
- Tradeoffs based on
  - Implementation Complexity
  - Perceived Threat Model
- Most tradeoffs are static policies
  - Partitioning users into groups with different levels of authentication
- How about a dynamic "*trust*" metric?
  - Trustworthy clients get better performance!

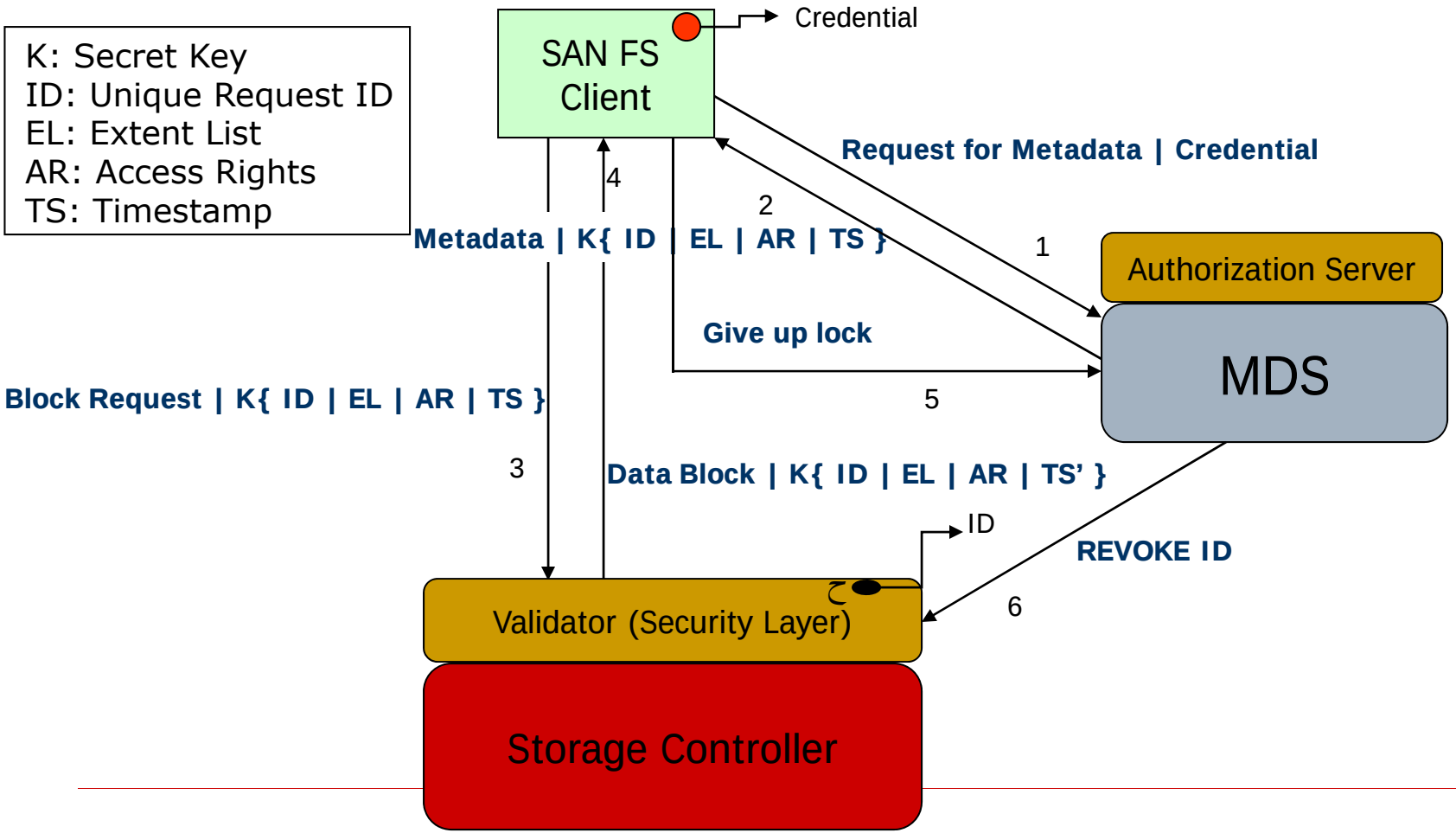
Trust and Trustworthiness

- Concept regularly practiced in P2P and e-commerce
- Provide differential treatment based on client behavior
  - Clients that are observed to behave correctly get better performance
  - **Dynamic evaluation**
- Two components
  - Trust Model: The "metric" of trust
  - Trust Distribution: The infrastructure for measuring and distributing trust information

Case Study: Direct Access SAN

- Direct Access SAN
  - Metadata Servers (MDS): Provide metadata information about files
  - Clients acquire metadata from MDS and access block storage ***directly***
- Block-level Security Solutions
  - Capability based mechanisms
  - MDS provides a capability to client which is validated at the storage controller
  - Requires cryptographic operations at storage

Example Secure Protocol



Differential Client Treatment

- *Good* Clients
  - A client that always accesses "correct" storage (appropriate blocks with appropriate read/write access perms)
  - A client that is authorized to access all storage (e.g. a compliance application)
- *Bad* Clients
  - Malicious client trying to access wrong blocks
  - Buggy application

- CAN be differentiated based on observed behavior
  - Ratio of correct transactions, e.g.

Trust Framework

- Good clients can get "trusted mode" access (TMA)
  - Storage trusts the client, does not validate capabilities and just provides requested access
  - Granted by the MDS based on the trust model and trust policy (e.g. Correct access > 99%)
  - To grant such access, MDS sends a message to storage to trust a particular credential
  - Revoked similar to a capability revocation

Enhanced Performance

- Need to ensure that *bad* clients do not get trusted mode access
  - Sufficiently strict trust model and trust policy

Trust Infrastructure

- Trust Model
  - Defines the metric of "trust"
  - Example of a binary trust model
    - Trust={0,1} 0 = not trusted, 1= trusted
  - Example of a continuous trust model
    - Trust = [0,1] 0 = least trusted 1=most

- Our case study
  - [0,1] Model
  - Trust Rating = $\begin{cases} 0, & \#tr < \Psi \\ (\#ctr/\#tr)^{1/\alpha}, & \#tr \geq \Psi \end{cases}$

#ctr: Number of correct transactions
#tr: Total number of transactions
α: Strictness parameter
Ψ: Threshold parameter

Trust Model (contd …)

- Pr(TMA) = Trust Rating = $\begin{cases} 0, & \#tr < \Psi \\ (\#ctr/\#tr)^{1/\alpha}, & \#tr \geq \Psi \end{cases}$
- Revocation of trusted mode access
  - Any instance of incorrect access, or
  - Whenever trust value drops
- Requires significant *good* history for gaining trusted mode access (Ψ)
- *Bad* behavior can be appropriately penalized (α)
- Extensions
  - Differential treatment based on data
    - Lower α for critical storage
  - Different levels of trusted mode access
    - Smaller security keys (32-bit encryption)

Trust Distribution

- Trust Ratings stored at MDS
  - MDS grants trusted mode access
- Statistics gathering
  - #tr, #ctr maintained as counters at storage controllers
  - During normal access, counters modified appropriately
  - During trusted mode access, counters modified through an auditing process
    - Requires logging!
- MDS gathers statistics from storage periodically

Trust References

- Karl Aberer's group at EPFL
  - PGrid
- Ling Liu's group at Georgia Tech
  - PeerTrust, TrustMe
- Hector Molina's group at Stanford
  - Eigenrep
- Muninder P. Singh's group at NCSU
- Virginia Lo's group at OGI

Conclusions and Future Work

- Presented a dynamic security/performance tradeoff mechanism
- Differential treatment of clients based on their observed behavior
- Measurement and policies of client trustworthiness
  - Dynamic and customizable trust model
- Empirical evaluation on standard benchmarks and threat models
- Possible enhancements in the trust distribution component of the infrastructure