

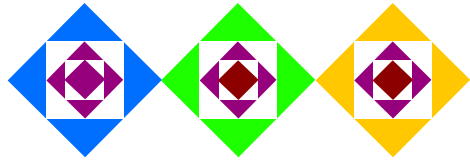
A Two Layered Approach for Securing an Object Store Network

A. Azagury, R. Canetti, M. Factor, S. Halevi, E. Henis,
D. Naor, N. Rinetzky, O. Rodeh and J. Satran

December 2002

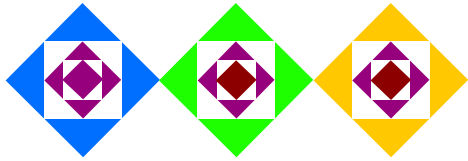
IBM Haifa Research Labs
and
IBM Watson Research Labs

Presenter: Dalit Naor



Talk Outline

- **What is an Object Store Network**
- **The Security Problem**
- **Related Work**
- **Trust assumptions**
- **Our Solution**
 - Two layered approach
 - Security definition
 - protocol
 - security
- **Further Work**



Object Store Network

■ What is an Object Store?

- Storage Device exposing 'object' interface
 - Create Object
 - Delete Object
 - Write(ObjectID, offset, length)
 - Read(ObjectID, offset, length)
 -
- Internal layout of object is transparent

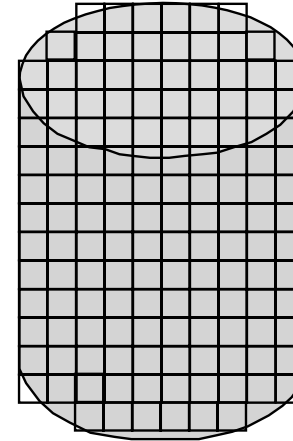
■ Intermediate granularity of abstraction

- larger than 'blocks'
- smaller than LUN (Logical Unit)

■ Architecture:

- Many Clients
- Many Server
- Most likely, also a *meta data server*
- Communication
 - IP (most likely)
 - Fiber Channel (FC)

Today's Block Device



Operations

read block
write block

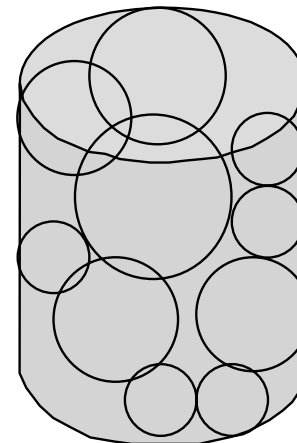
Security

Weak
Full disk

Allocation

External

Object Store



Operations

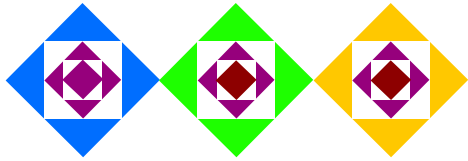
read object offset
write object offset
create object
delete object

Security

Strong
Per Object

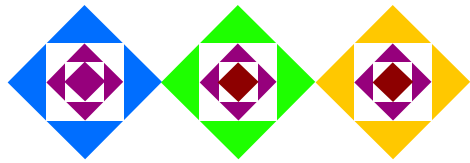
Allocation

Local



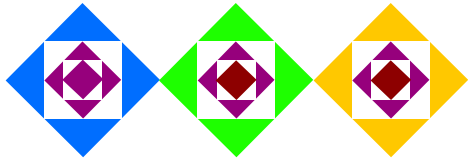
ObS Security Objectives

- **Enforcing Access Control at Object Granularity**
 - Application specific AC
 - Policy itself is 'external', provides flexibility
 - Increased protection/security at level of objects rather than whole LUs
 - Allow shared access to storage without giving hosts access to all data on volume
 - Allow non-trusted clients to sit in the SAN
 - Threats
 - Unauthorized operations by a principal
 - Illegal use by a principal of expired/revoked permissions
- **Protecting against network attacks**
 - Threats like:
 - Principal masquerades as a different principal or as an object store
 - Eavesdropping, message modification, message replay
- **Keep acceptable performance:**
 - computation and message bandwidth mainly on critical path
- **Simple administration and deployment**



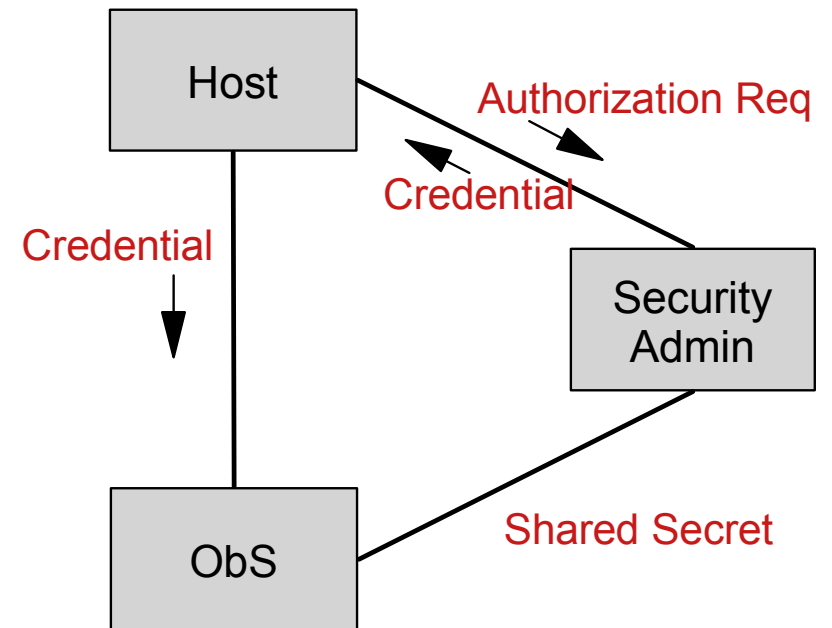
Related Work

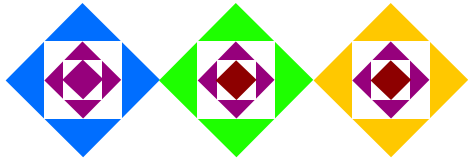
- **SAN Security gains a lot of attention**
 - Authentication
 - Encryption
- **Encryption systems**
 - CFS (Cryptographic File System), 1993
- **Distributed FS**
 - AFS , Kerberos-like
- **NADS (Network Attached Distributed Storage)**
 - Most comprehensive, Security by H. Gobiuff 1999
 - Credential based
 - Tie network security with credential
 - Key management
- **SCARED , 2000**
 - Mutual Authentication
 - Simple key management
 - MAC based
- **SNAD , 2002**
 - Encryption



ObS Security - Solution

- **Credential Based**
 - All operations are secured by a credential
- **Security achieved by cooperation of:**
 - Security Administrator (e.g. embedded within the File Manager) - authenticates, authorizes and generates credentials.
 - ObS - validates credential that a host presents.
- **Credential is cryptographically hardened**
 - OSD and Admin share a secret key
 - The shared key is periodically refreshed





Trust Assumptions

■ Servers

- data integrity
- follow protocol
- machine can not be controlled by an adversary

■ Admin

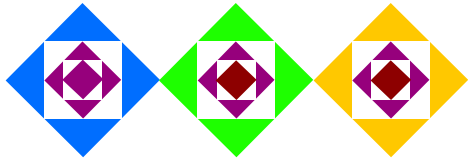
- highly trusted
- stores lone-lived keys, computes AC correctly
- machine can not be controlled by an adversary

■ Client

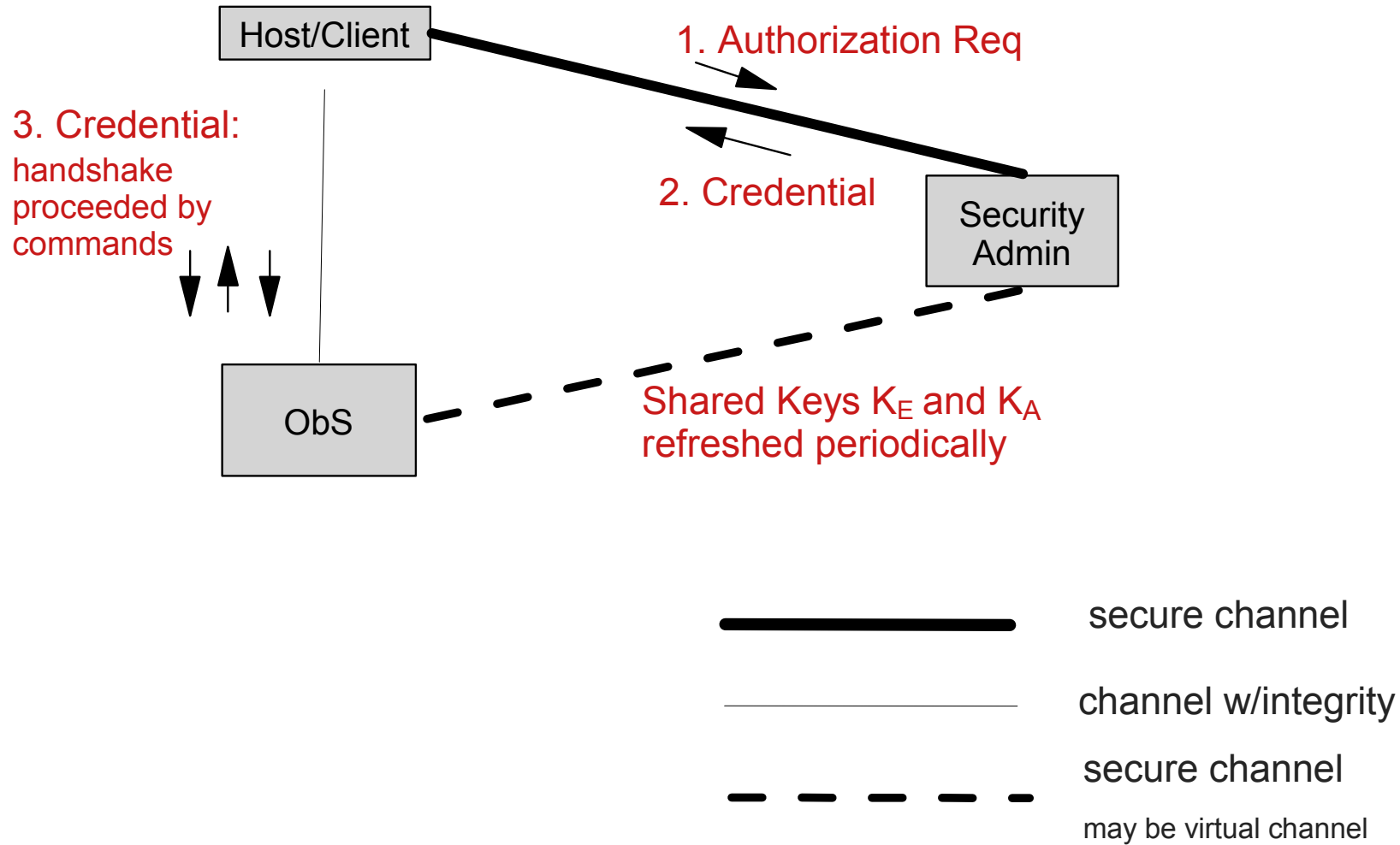
- malicious software
- hostile machine

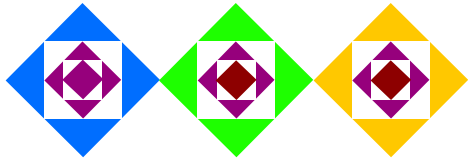
■ Communication Links

- Client-Server channel is authenticated
 - no secrecy, adversary can observe traffic
 - adversary may use links that are available to the client under control
- Other channels are encrypted
 - adversary may only do traffic analysis and block messages



Security Flow and Channel Requirements

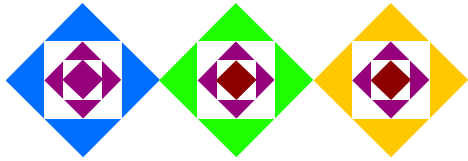




Clients, Machines and Links

What if several clients on a single machine?

- **OS effectively separates between clients .**
- **If OS is compromised**
 - all clients on this machines are compromised.
- **Not an issue for server**
- **Authentication is done only by Admin**
 - Client-Server links are *authenticated but anonymous*
 - all messages arrive from same party
 - no modifications



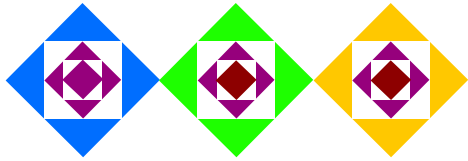
Our Contributions

1. Two layered approach

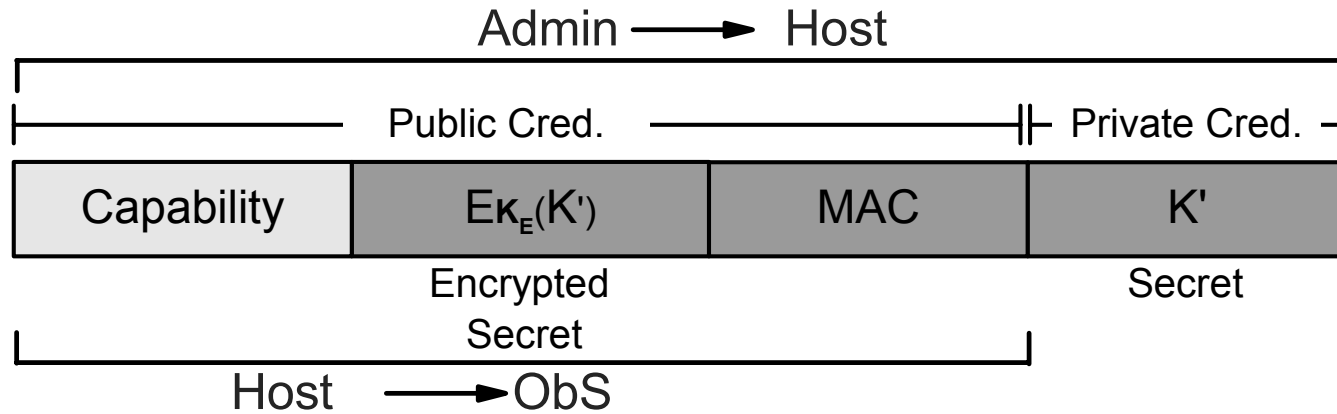
- Authorization
 - An authorization protocol with standard cryptographic building-blocks
- Network security
 - Standard off-the-shelf mechanisms for transport layer security: authentication, message-integrity, and encryption
 - **IP**: IPsec as the secure transport layer
 - **Fibre-Channel**: assume protected network or any suture security solutions
- Rationale: leverage existing infrastructure for standard layer (network security)
 - Avoid duplication of function
 - Better performance
 - Leverage work of others
 - Easier to convince protocol is correct
 - Cheaper to develop

2. Security Definition

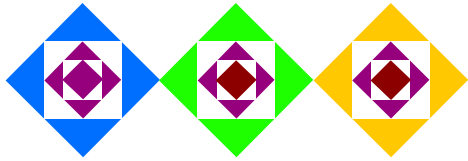
- A run of protocol with an adversary. Adversary may:
 - control client machine(s)
 - use communication channels as described
- Adversary 'wins' if gets a permission on an object it otherwise does not control
- *Protocol is secure if a feasible adversary wins with a negligible probability.*



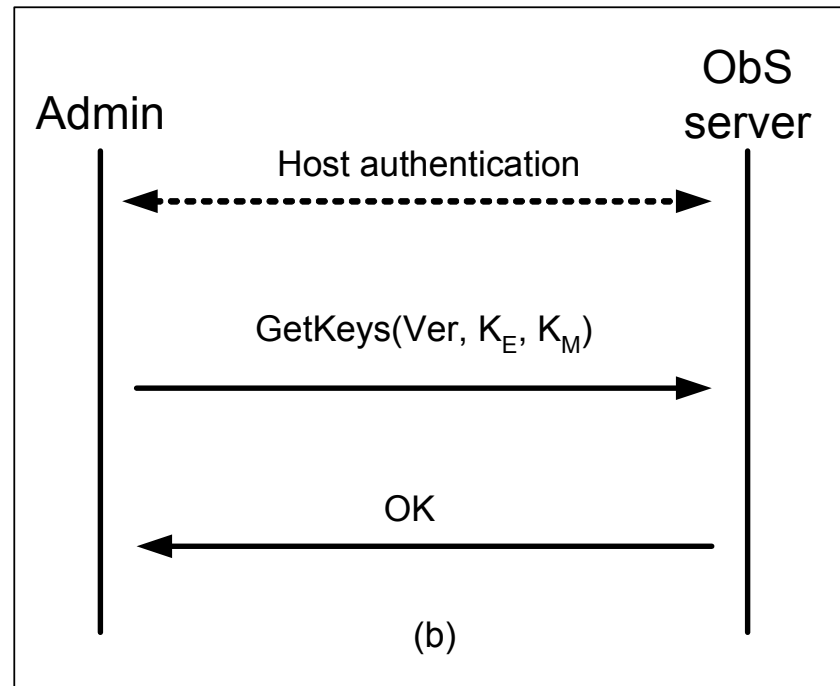
Credential Structure

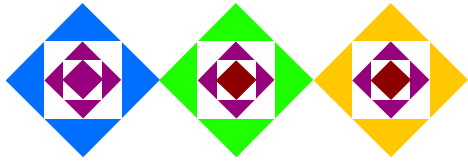


- **Capability**
 - Operations that the credential entitles.
- **Encrypted Secret**
 - A Secret generated by the Admin
 - A different secret for every credential
 - Encrypted with a key Admin shares with the ObS
- **MAC**
 - Over previous fields to ensure integrity
- **Secret**
 - The (un-encrypted) secret
 - Used by the ObS to verify that the host got credential from the Admin.

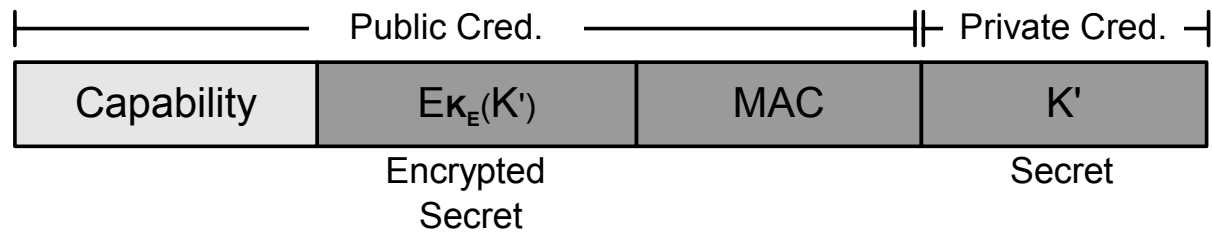
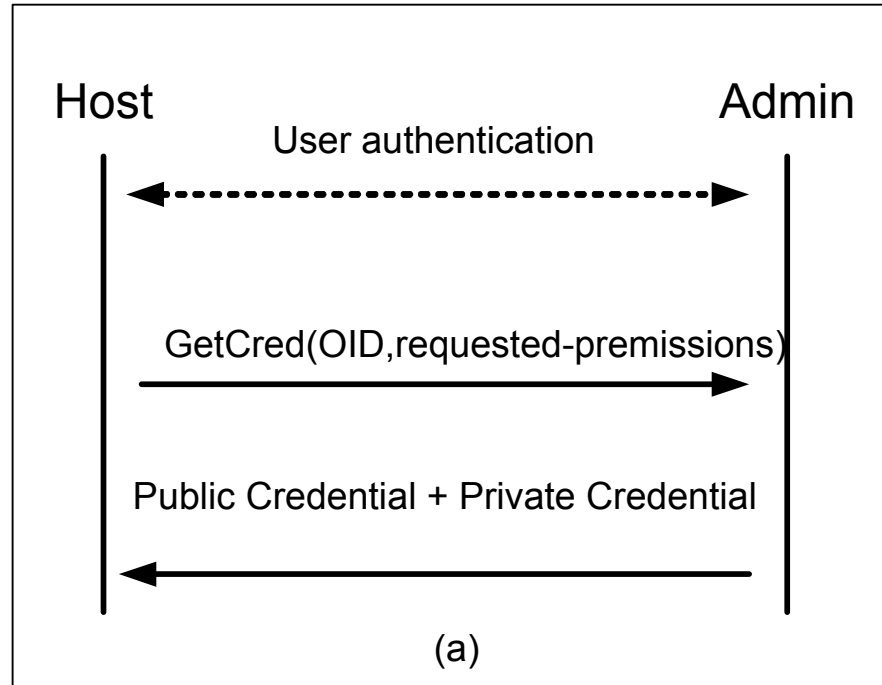


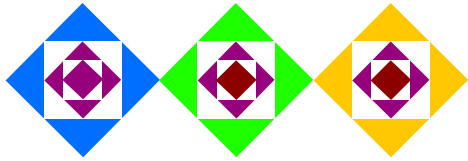
Protocol Between ObS and Admin



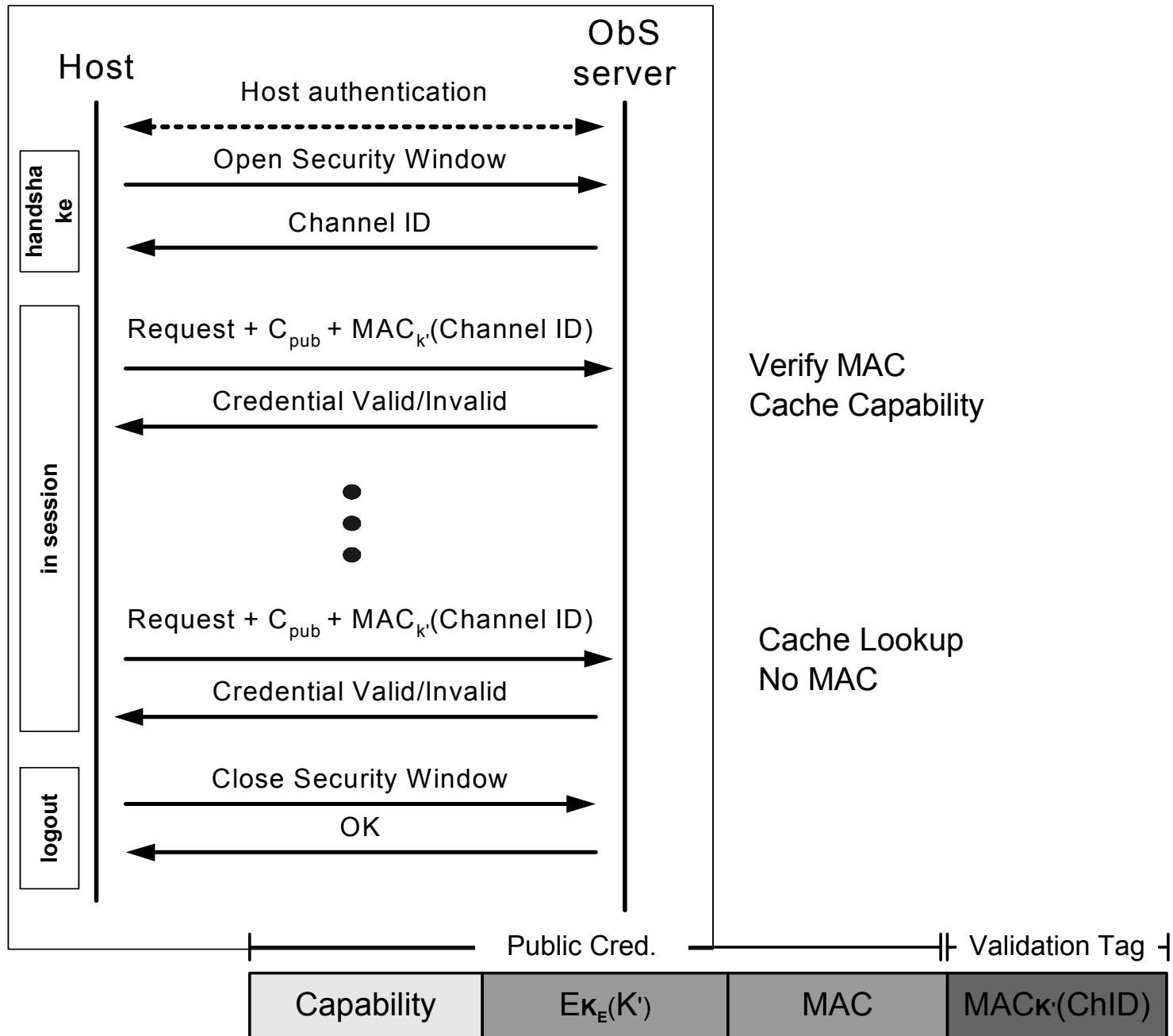


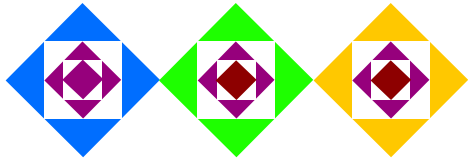
Protocol Between Host and Admin





Protocol Between Host and ObS





Security Flow

■ Preliminary

- Client authenticates to security Admin, e.g.,
 - Third party: Kerberos in AFS, VeriSign ...
- ObS authenticates to security Admin over a secure channel

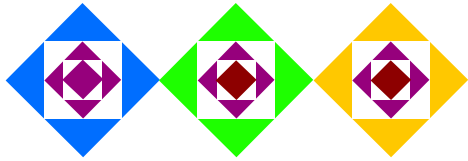
■ Basic Flow

- Client presents authenticator and requests credential from Admin
- Send requests to ObS with appropriate credential
- ObS verifies credential and performs operation
- Credential verification is fast
 - involves only symmetric key operations

■ To improve CPU and bandwidth utilization we use credential caches

- At the client: saves accesses to the security Admin
- At the ObS per client: saves repeating verification of credentials

■ Summary: the critical path remains short



Proof Of Security

- **Claim:**

- An adversary can not access objects other than those that are permissible to clients under its control.

- **Method of proof:**

- a reduction to the security of the underlying primitives E_K and MAC_K .
- Main Claim:

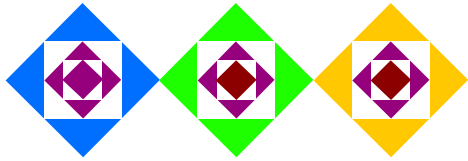
If an adversary wins with probability ε , then at least (i) or (ii) holds:

(i) \exists an attacker that breaks the MAC with probability $> \min(\varepsilon / 3M, \varepsilon / 3N)$

(ii) \exists an attacker that distinguished the encryption E from a random permutation with advantage $> \varepsilon / 2N$

M - # authentication key exchanges with Admin.

N - # credentials the Admin generated.



Future Work

- **Performance analysis**
 - Currently, within the Antara Object Store prototype.
 - Preliminary results:
 - cache of credentials is essential
 - cryptographic verifications is very fast
- **Fibre Channel - alternatives**
 - Implement a secure channel
 - stripped version of IPSec