

Recent-Secure Authentication: Enforcing Revocation in Distributed Systems

Stuart G. Stubblebine

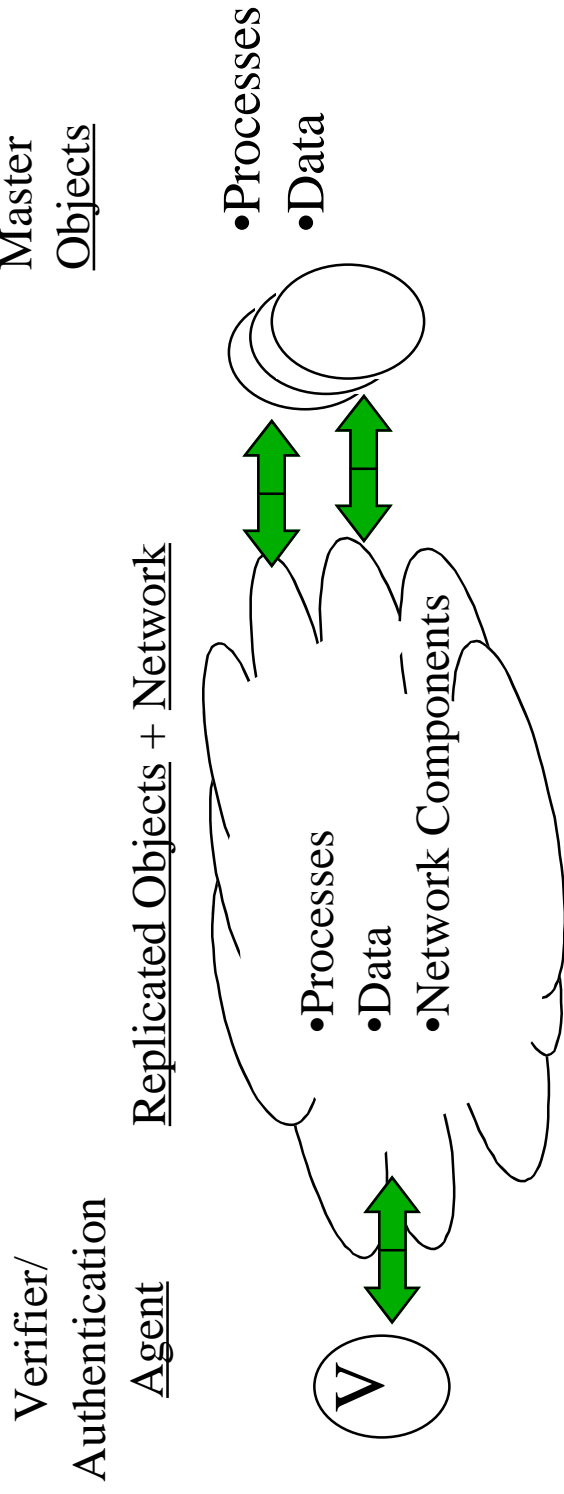
AT&T Bell Labs

May 1995

Topics

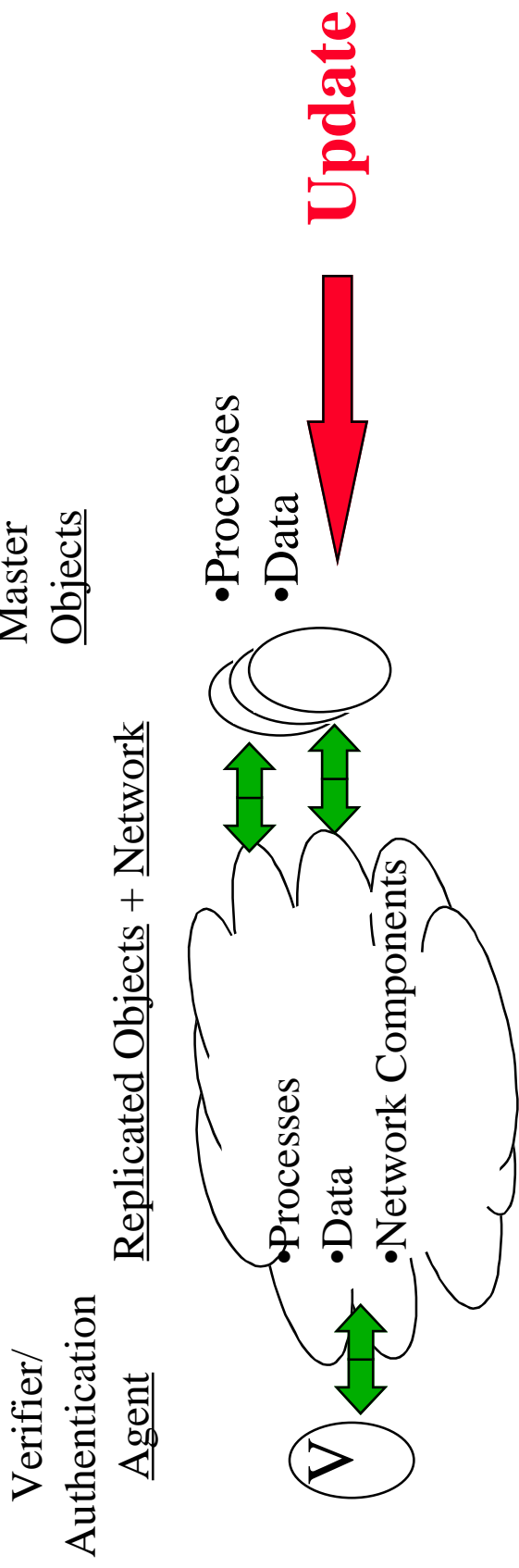
- Revocation in Large Distributed Systems:
Requirements
- Analysis Method and Design Techniques
- Applied to Trusted Third Party Revocation
- Summary

Channel Authentication in Large Distributed Systems



- Verifier needs to establish: Who said something?
- Verifier uses a variety of /multiple distributed objects.

Revocation of Channels in Large Distributed Systems

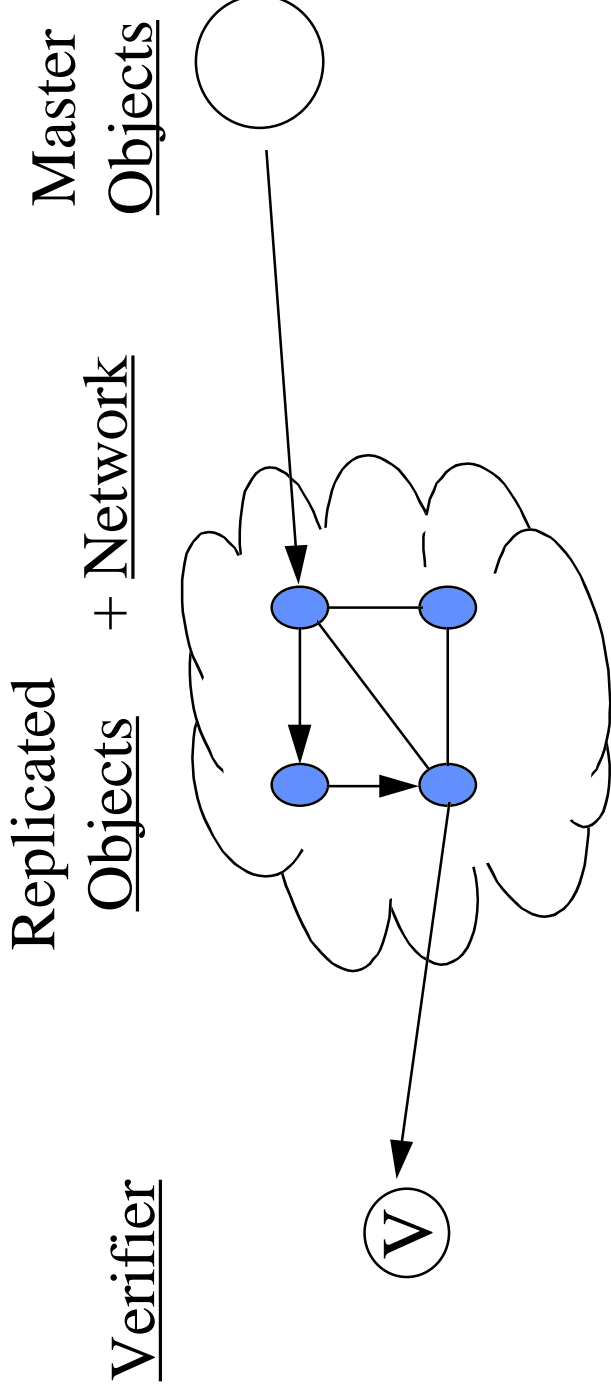


- Goal: Halt authentication based on outdated information.
- Network & Replicated Objects introduce delays.

Difficulty of Revocation in Large Distributed Systems

- **FACTORS:**
 - Size
 - Trust
 - Security
 - Distributed and Faulty
 - Temporal Dynamics
 - Latency

Latency



- Communication Latency is Inherent to Distributed Systems
 - Verifiers can not have perfect knowledge of authentication/authorization information
- Replicated objects increase the delay

Requirements for a Revocation Mechanism & Service

- Definite and Bounded Delay
- Available and Recent
- Adjustable

Given a Compromise of the Revocation Service ...

- Bounded delay for Recovery
- Contain the spread of a compromise

Definite and Bounded Delay for Revocation (Fail-Safe Property)

- Broadcasting Revocation Lists or “Emergency Updates” alone is Insufficient
 - network is assumed unreliable
 - adversary can block revocation updates at will.

- Require Fail-Safe Operation:

Verifier is ***Guaranteed***

NOT to use

Outdated Information

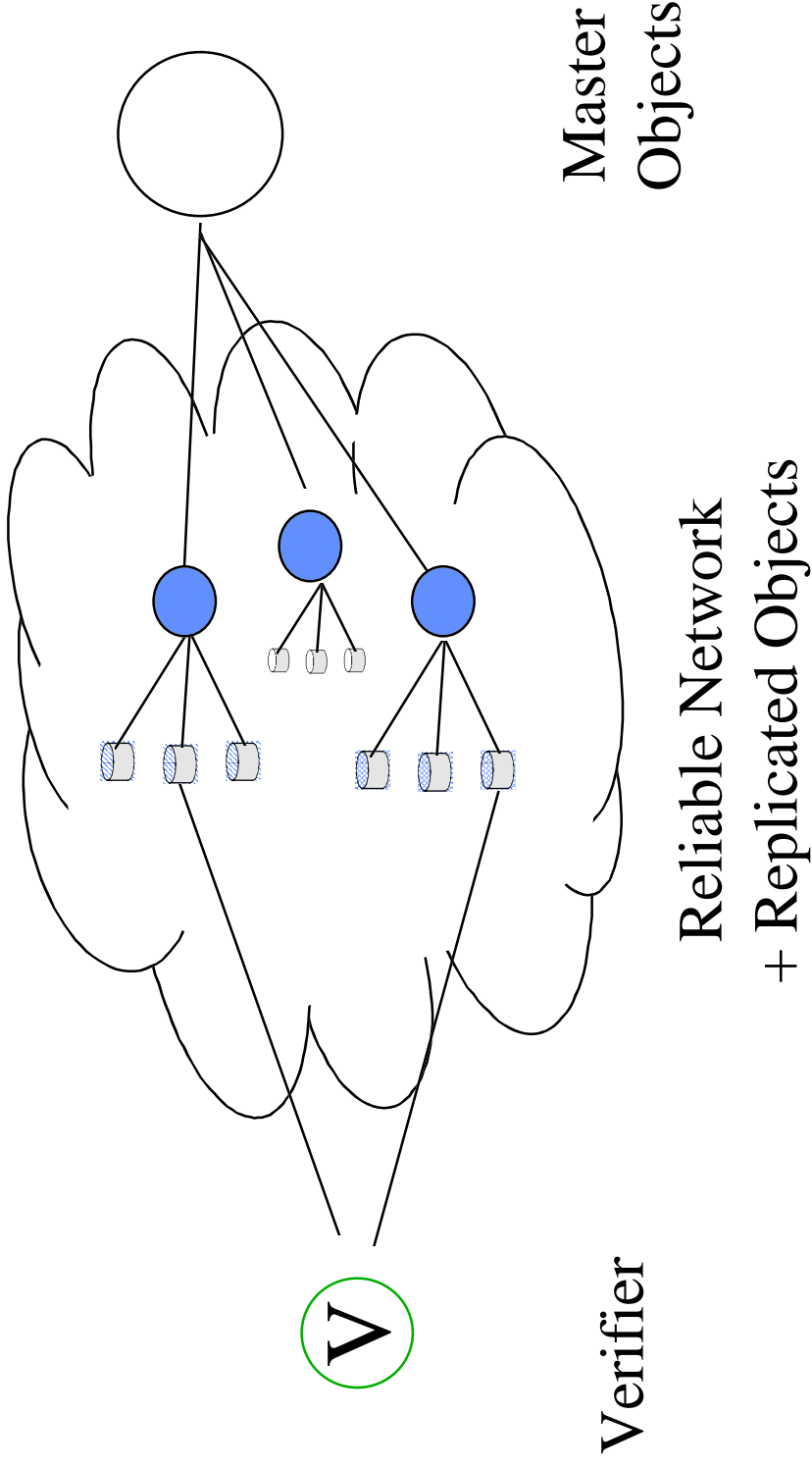
Update to Master

Time:

t

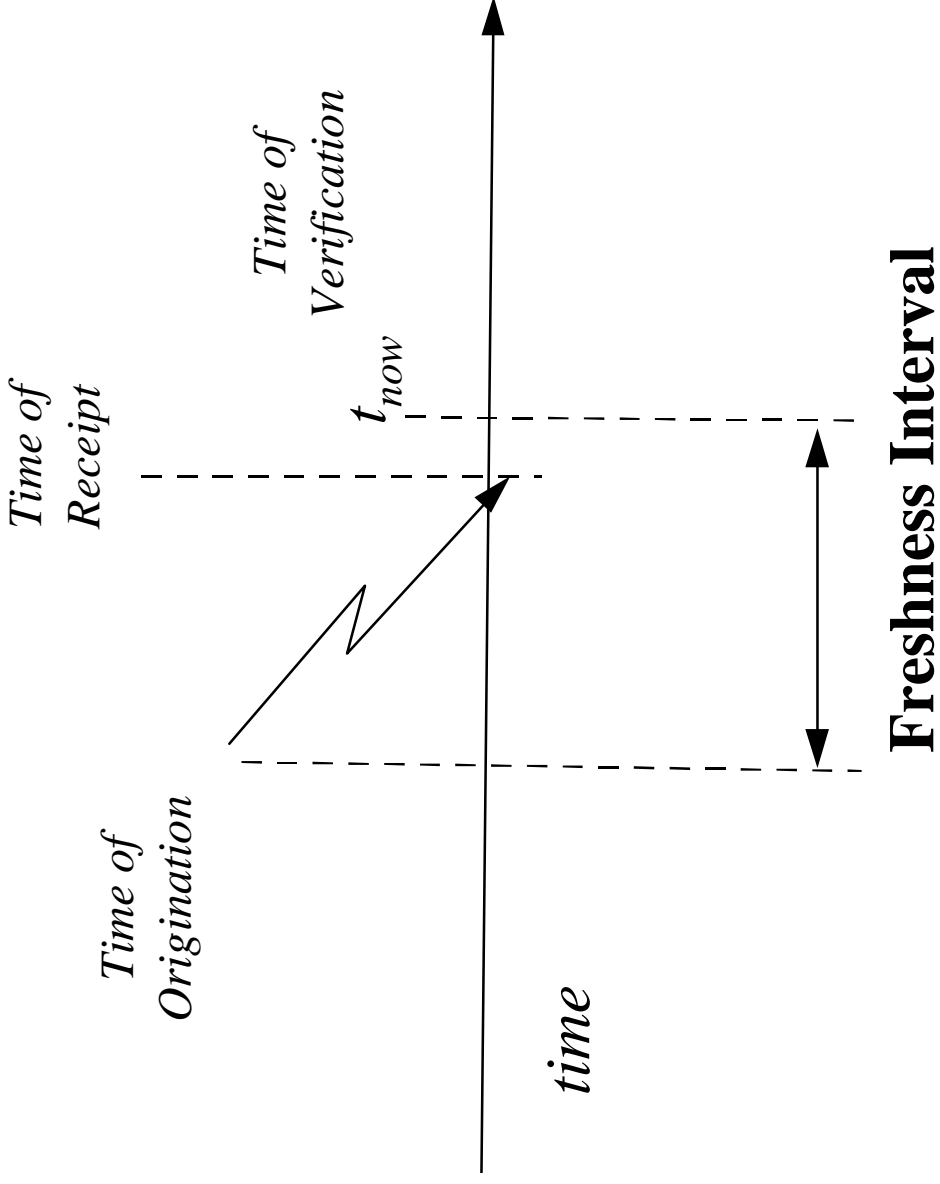
t + max_delay

Available



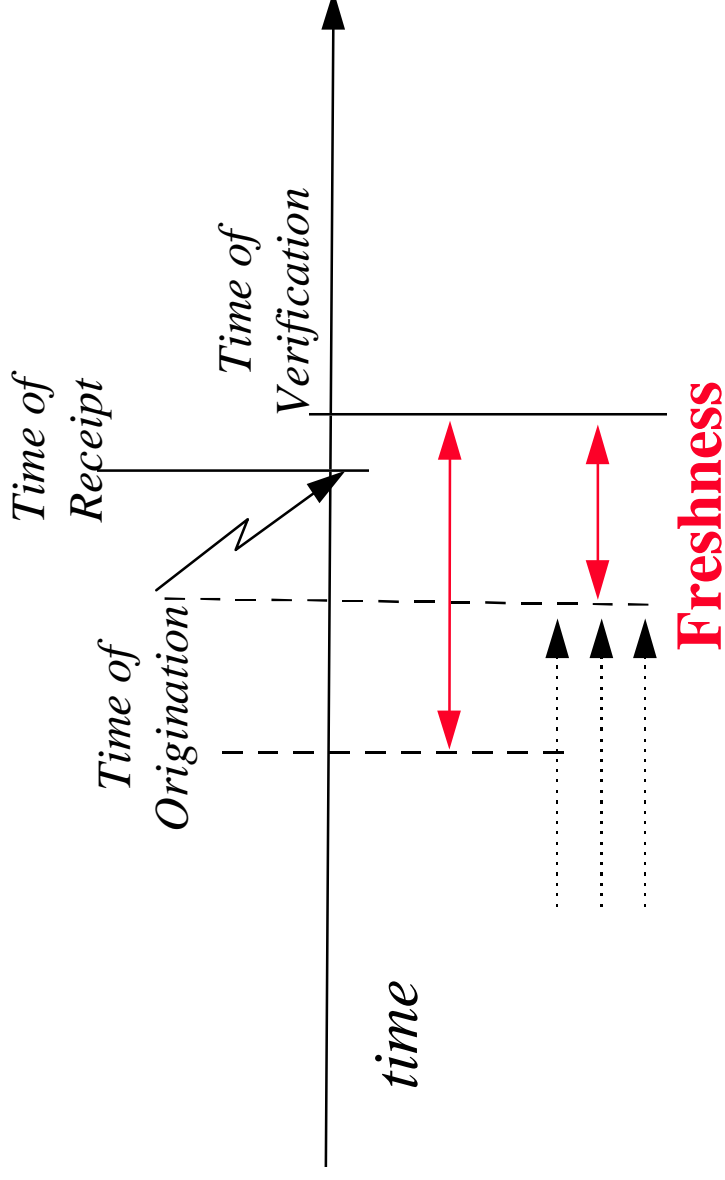
•Path Diversity

Recent Information



Adjustable: Architecture supports a range of

freshness intervals



- Low value transactions may use stale information
- High value transactions may require fresh information

Method and Techniques

- Framework for Analysis of Recent-Secure Channel Authentication
 - Built on secure channels [LABW, ABLP]
- Approach:
 - Specify Freshness (Policy) Constraints
 - Analyze Authentication against Freshness Policies
- Techniques:
 - Delegation with freshness constraints (not explicit expiration times!)
 - Pointers/Indirection within long-lived certificates

Review: “Calculus for access control in distributed systems”

[ABLP, LABW]

- Named Principals: Bob, Bob as Manager, CA, Department X
- Channel Principals: K_{CA}
- Statements:
 - “says”: K_{CA} **says** s ; - “Speaksfor”: $K_{CA} \Rightarrow CA, CA \Rightarrow Bob$
- Principal Operations:
 - quoting: $A \mid B$ **says** $s \equiv A$ **says** B **says** s - conjunction: $(A \text{ and } B)$
- Axioms:
 1. $(A \Rightarrow B) \supset ((A \text{ says } s) \text{ implies } (B \text{ says } s))$
 2. $(B \text{ says } (A \Rightarrow B)) \supset (A \Rightarrow B)$
- Example. *Given:* $CA \Rightarrow Bob, K_{CA} \Rightarrow CA$
Receive: K_{CA} **says** $(K_{Bob} \Rightarrow Bob)$ (certificate)
Deduce: CA **says** $(K_{Bob} \Rightarrow Bob)$ from rule 1
Bob says $(K_{Bob} \Rightarrow Bob)$ from rule 1
 $K_{Bob} \Rightarrow Bob$ from rule 2

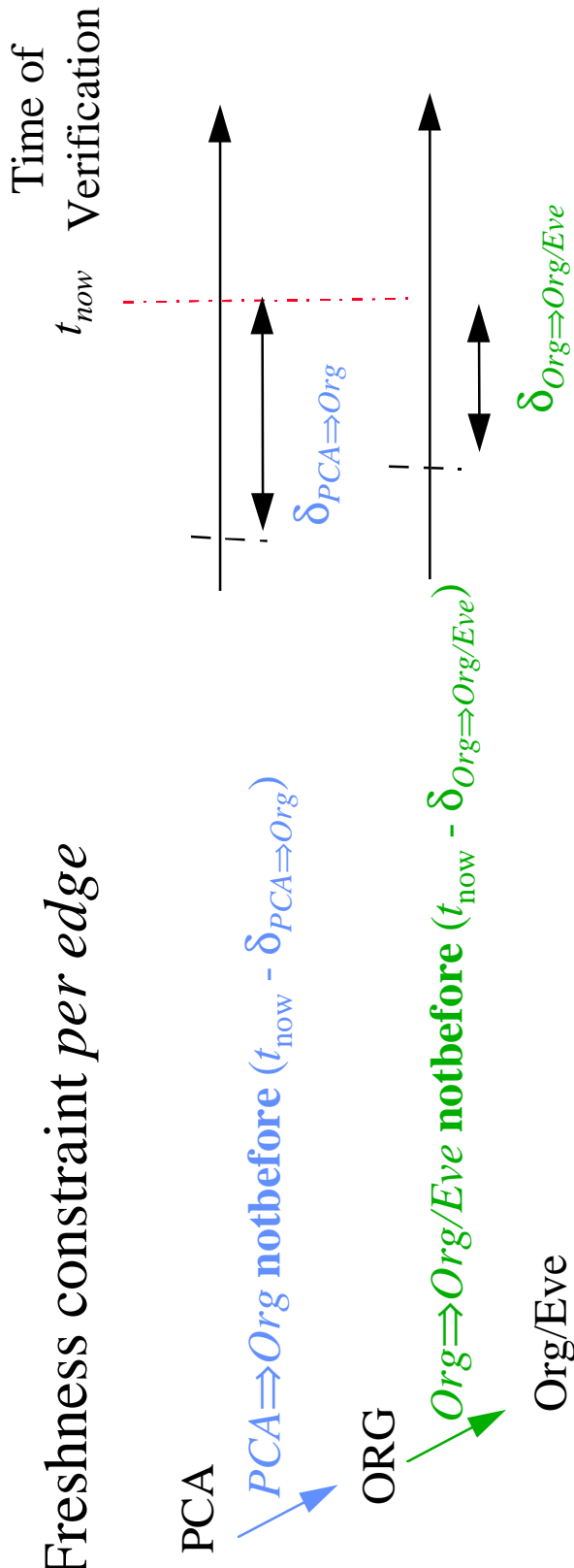
Extensions: Qualifying Statements with Time

- Validity Constraints: $K_{CA} \Rightarrow CA$ **notbefore** t_1 **notafter** t_2
 - Holds on the closed interval $[t_1, t_2]$
- Time a principal “says” a statement, s : K_{CA} **says** s **at** t
- Example.
 - Certificate: $\{B, K_B, \text{notbefore } t_1, \text{notafter } t_2, \text{timestamp}\}^{K^{-1} CA}$
 - *Formalism:*

K_{CA} **says** ($K_B \Rightarrow B$ **notbefore** t_1 **notafter** t_2) **at** *timestamp*

Specifying Freshness Policies/Constraints

- Freshness constraint *per edge*



- Context Dependent:

- Per Certification Authority, HW/SW assurance, Transaction Amount, position in certificate path, etc.
- Ex.: $\delta_{CA \Rightarrow CA/Eve} = 30 \text{ min.}$ Transactions over \$100
- Revocation bound: Maximum delay on all paths to a named principal.

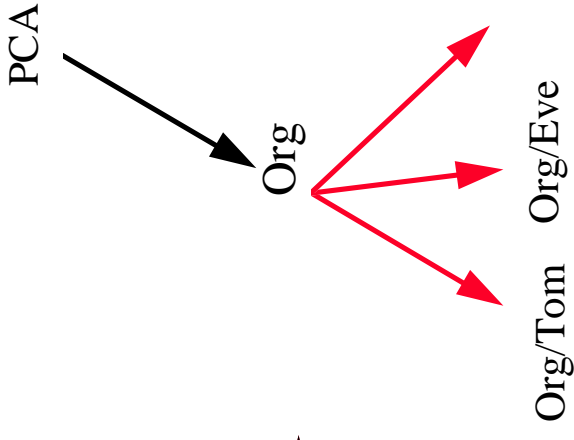
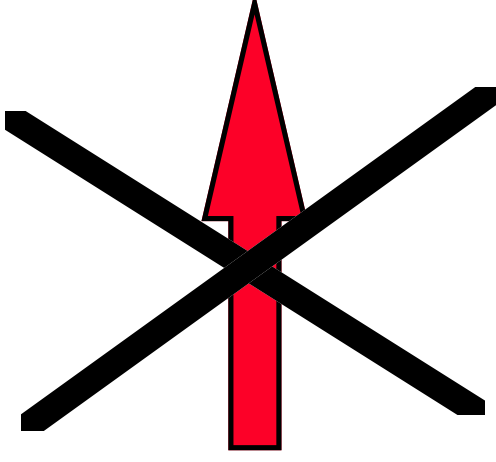
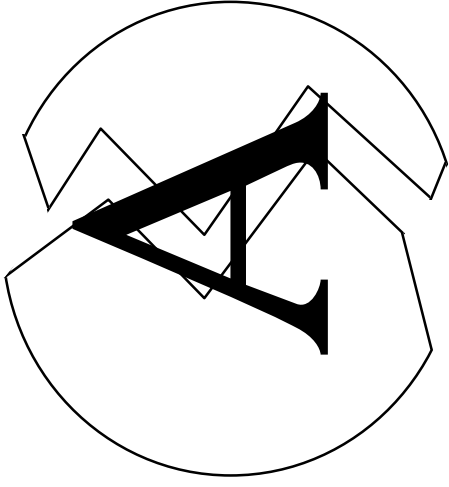
Axioms

- (P6): $(A \Rightarrow B \text{ notbefore } t_1 \text{ notafter } t_2) \supset$
 $((t_1 \leq t_3) \text{ and } (t_4 \leq t_2))$ implies
 $A \Rightarrow B \text{ notbefore } t_3 \text{ notafter } t_4$
- (P7): $(A \Rightarrow B \text{ notbefore } t_1 \text{ notafter } t_2) \supset$
 $(A \text{ says } s \text{ at } t_3) \text{ and } (t_1 \leq t_{\text{now}}, t_3 \leq t_2)$ implies
 $(B \text{ says } s \text{ at } t_3)$
- (P8): $(A \text{ says } (B \Rightarrow A \text{ notbefore } t_1 \text{ notafter } t_2) \text{ at } t_3) \supset$
 $(B \Rightarrow A \text{ notbefore } t_1 \text{ notafter } t_2)$

Recent-Secure Channels for a Link of a Certificate Path

- A variety of design approaches are possible for each link of a certificate path.
- Examples:
 - Identification Certificate + Timestamped Revocation Certificate
 - revocation certificate asserts the validity of a referenced identification certificate at the time the statement is made.
 - Identification Certificate with timestamp
 - X.509 standard does not have this option.

Requirements for Recovery from a Compromised Revocation Service: Contained



**Compromised
Revocation
Authority**

**Generation
of unauthorized
Certificates**

Joint Authority to Satisfy Containment Objective

- Off-line Certification Authority (CA) + On-line Entity (O)
 - K_{CA} says $((K_B \Rightarrow B \text{ notbefore } t_1 \text{ notafter } t_2)$ and $O|K_B \Rightarrow B \text{ notbefore } t_3 \text{ notafter } t_4)$)
 - $O|K_B$ says $(O|K_B \Rightarrow B \text{ notafter } t_5)$ (explicit expiration time)
 - Derive: $K_B \Rightarrow B \text{ notbefore } t_6 \text{ notafter } t_7$
- Benefit:
 - Adversary can't issue new identification certificates if On-line entity is compromised.
 - CA can be made more secure since it is off-line.
- Limitation: Given a compromise of the On-line Entity, *revocation can be can be delayed well beyond the intended freshness constraint* on the identification certificate.
 - Ex. : Adversary issues revocation certificate with an arbitrary expiration time t_5 . The validity of the revocation authority is only checked when the revocation certificate is *first* obtained.

Compromised Revocation Service: *Bounded Delay for Recovery*

Revocation failures due to a compromised revocation service should be reasonably bounded in time.

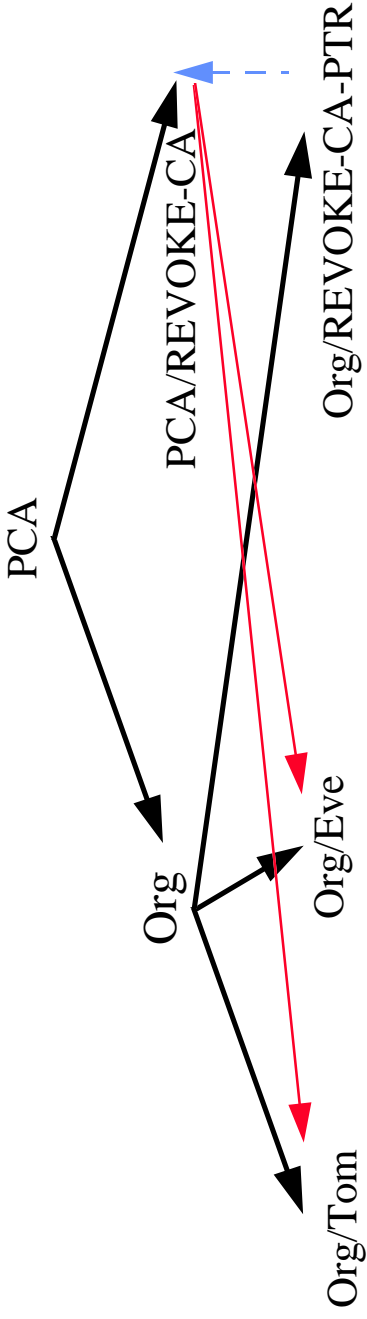
Technique for Bounded Delay for Recovery: Freshness-Restricted Delegation

- Specify the *freshness constraint* (of the trusted link) with the delegation of revocation authority.
 - Timestamped revocation certificate NOT certificate with expiration time.
- Certificate & Formal Description:
 $\{Org/Eve, K_{Org/Eve}, \mathbf{notbefore } t_1, \mathbf{notafter } t_2, \text{Revocation Authority} = Org/REVOKE-CA-PTR, \text{with freshness } \delta_{Org \Rightarrow Org/Eve} = 30 \text{ minutes}\} K^{-1}_{Org}$
 $K_{Org} \text{ says } ((K_{Org/Eve} \Rightarrow Org/Eve \mathbf{notbefore } t_1 \mathbf{notafter } t_2) \text{ and } (Org/REVOKE-CA-PTR | K_{Org/Eve} \Rightarrow Org/Eve \mathbf{notbefore } (t_{now} - \delta_{Org \Rightarrow Org/Eve})))$
- Delay bounded to freshness constraint on certificate (-not well past it!)
 - Verifier knows the freshness constraint on the certificate path
 - Verifier checks the validity of the revocation authority each time a revocation certificate is obtained.

Example: Trusted Third-Party Revocation

- **Objective:**
 - *Improve performance* by consolidating revocation authority functions.
 - *Minimize the trust* required of a revocation service
- **Approach:**
 - *Freshness-restricted delegation* to a third-party revocation entity
 - *Indirection* within long-lived certificates for flexibility

Certification Topography and Description



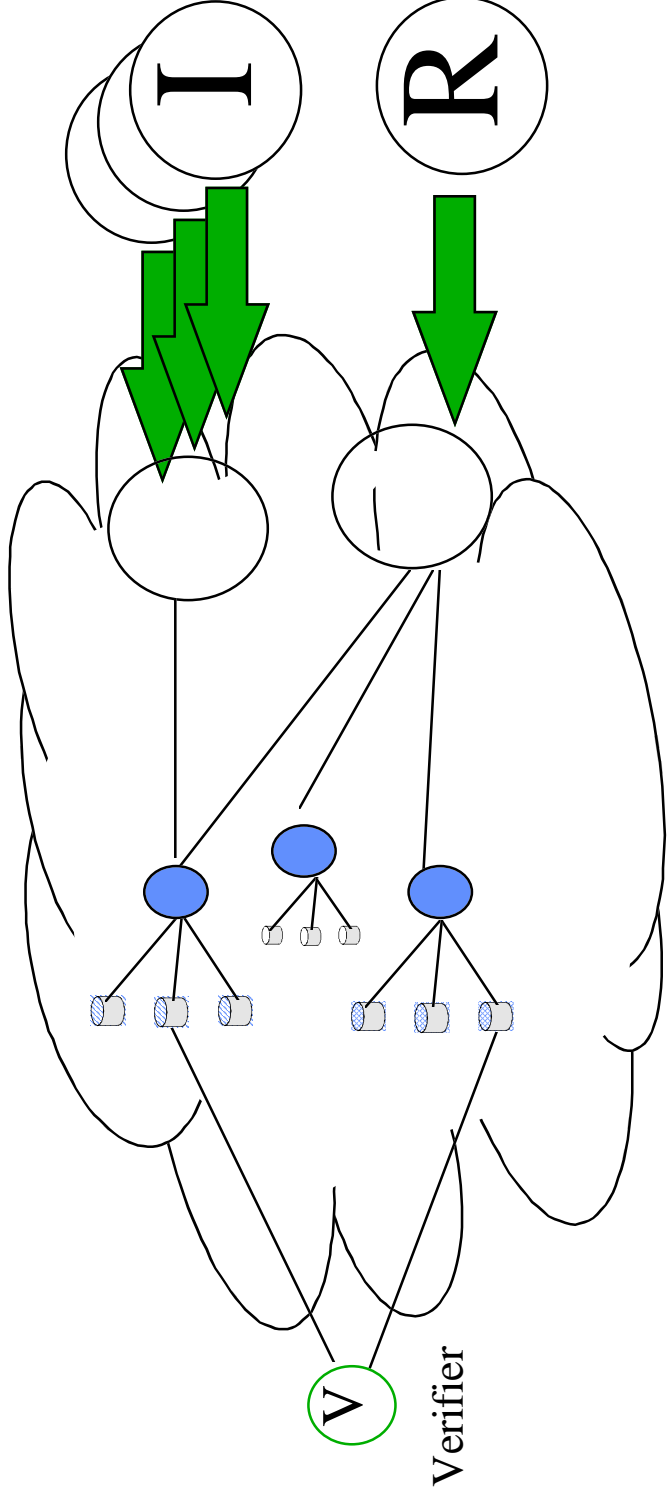
Long-Term Identification K_{Org} says $((K_{Org/Eve} \Rightarrow Org/Eve$ notbefore t_1 notafter $t_2)$ and
 Certificate: \longrightarrow $(Org/REVOKE-CA-PTR | K_{Org/Eve} \Rightarrow Org/Eve$

notbefore $(t_{now} - \delta_{Org \Rightarrow Org/Eve})$)

Medium-Term Delegation K_{Org} says $(PCA/REVOKE-CA \Rightarrow Org/REVOKE-CA-PTR$
 Certificate: \dashrightarrow **notbefore t_3 notafter $t_4)$**

Timestamped Revocation $K_{PCA/REVOKE-CA} / K_{Org}$ says $(K_{Org/Eve} \Rightarrow Org/Eve$
 Certificate: \longrightarrow **notbefore t_5 notafter $(t_6 + \delta_{Org \Rightarrow Org/Eve})$ at t_6**

Distribution Architecture



- Secure: *Uni-direction* Network Connectivity (for I & R)
 - separate, highly available, bi-directional channels used for management (e.g., between agents for I & R).
- Replicated objects provide *bounded delay guarantees* for updates
- Verifiers subscribe* to replicated objects according to recent-secure authentication requirements.

Summary

- **Formalizing Recent-Secure Authentication**
 - *Bound delay for fail-safe Revocation*
 - important metric for differentiating between designs
 - *Characterize degrees of verifier protection*
 - step towards decentralized risk management
- **Method for Analysis of Recent-Secure Authentication**
 - Analysis of designs for satisfying recent-secure channel authentication objectives
- **Trusted Third-Party Revocation Design & Techniques:**
 - *Delegations with freshness constraints*
 - Towards consolidation of revocation authorities
 - *Indirection in long-term certificates*
 - Flexibility in changing trusted third-party revocation authority.