I·C·S
The Institute for Cyber Security

C·SPECC
Center for Security and Privacy Enhanced Cloud Computing

Refresh Instead of Revoke Enhances Safety and Availability: A Formal Analysis

Mehrnoosh Shakarami
Ravi Sandhu

Institute for Cyber Security (ICS)
Center for Security and Privacy Enhanced Cloud Computing (C-SPECC)
Department of Computer Science
University of Texas at San Antonio

33rd Annual IFIP WG 11.3
Conference on Data and Applications Security and Privacy (DBSec'19)
Charleston, SC, USA - July 15-17, 2019

World-Leading Research with Real-World Impact!

UTSA Computer Science

1
Introduction & Motivation
- What is Attribute Based Access Control?
- Why I should care about consistency problem?

2
System Model and Assumptions
- Refresh vs. Revocation
- System assumptions and preliminaries

3
Consistency Levels Formal Characterization
- Proposed levels in a glance
- Level details and properties

4
Discussion, Conclusion and Future Work
- Limitations and Practical Issues
- What has been done? What to do next?

Introduction

- Access control regulates access to protected resources in the system with respect to the policy.

SUBJECT

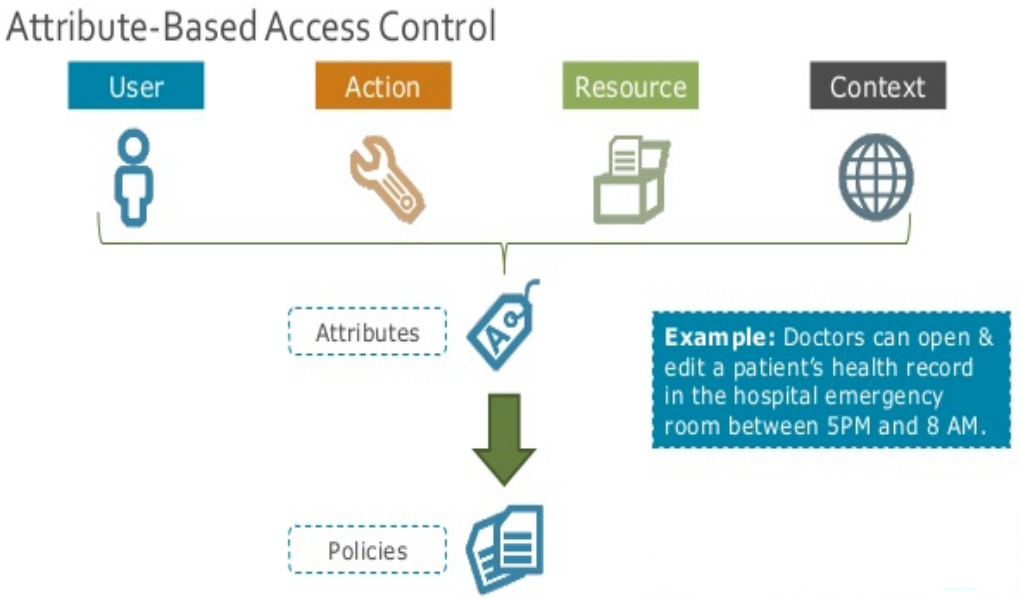
Generally an individual, process, or device causing information to flow among objects or change to the system state.

OBJECT

System-related protected entity (e.g., devices, files, records, tables, processes, programs, domains) containing or receiving information.

Policy

A set of rules which regulates access of subjects to protected objects in the system.

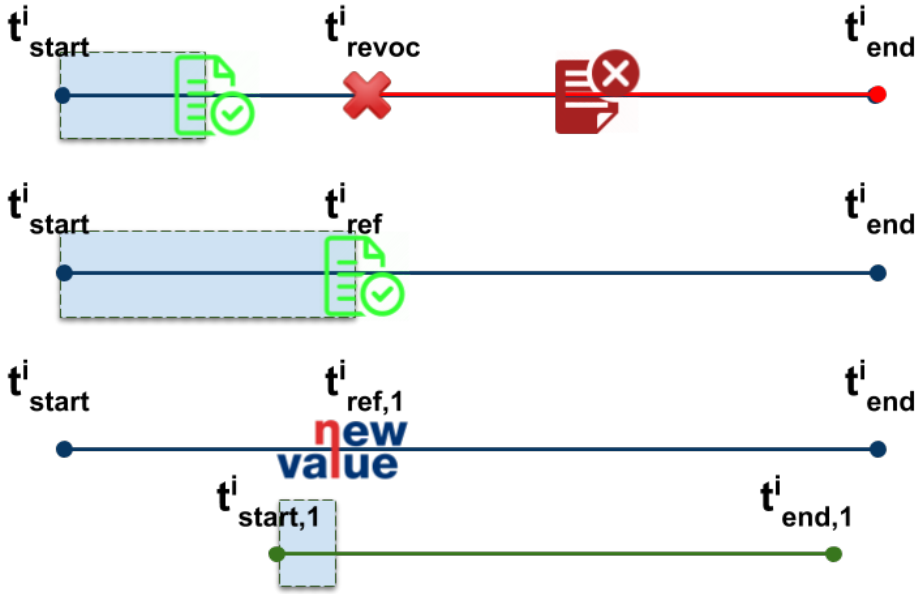


Consistency Problem

- Consistency Problem: When multiple attributes are involved, consistency problem results in granting access when it should be denied (safety violation) or denying access when it should be granted (availability violation), due to following reasons:
  - Asynchronous nature of distributed systems
  - Cached values of attributes
  - Network and system failures
  - Incremental assembly of subject attributes
  - Differing validity periods for subject attribute values

Refresh vs. Revocation

- Assuming an ABAC model in place, we define our consistency notions.
- The value of a subject attribute is referred to as a *credential* with specific lifetime, which can be revoked. But in refresh scenario we can get possible new values.



	Revocation	Refresh
First Check	Valid	Still-Good
Second Check	Invalid	Invalid

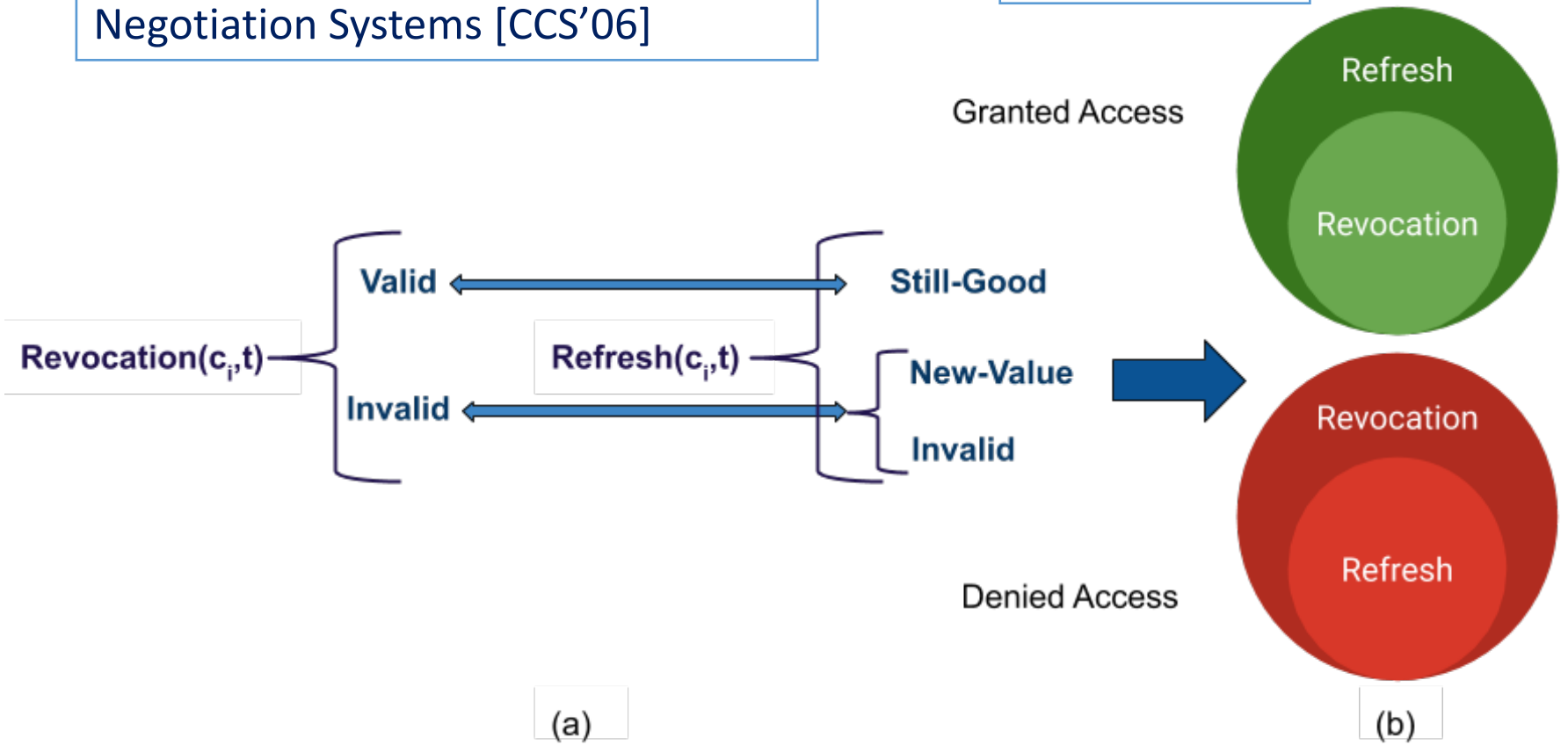
	Revocation	Refresh
First Check	Valid	Still-Good

	Revocation	Refresh
First Check	Invalid	New-Value

Refresh vs. Revocation

Pioneered by Lee-Winslett in Trust Negotiation Systems [CCS'06]

Our Approach



$$Refresh : C \times T \rightarrow \{Invalid, Still-Good, New-Value\}$$
$$Revocation : C \times T \rightarrow \{Invalid, Valid\}$$

Preliminaries

- Table of Symbols used during presentation:

Symbol	Meaning
t_{req}	request time
t_d	decision time
c_i	i-th credential
t^i_{revoc}	actual revocation time of c_i (the AA always knows this time)
$t^i_{ref,k}$	time of k-th refresh of c_i
$t^i_{start,k}$	attribute start time of c_i after k-th refresh
$t^i_{end,k}$	attribute expiration time of ci after k-th refresh
$kmax(t)$	latest refresh of c_i before time t (c_i is determined by context)

Preliminaries (cont.)

- Satisfactory Values: We define an attribute to be satisfactory if and only if its value fulfills the policy conditions.

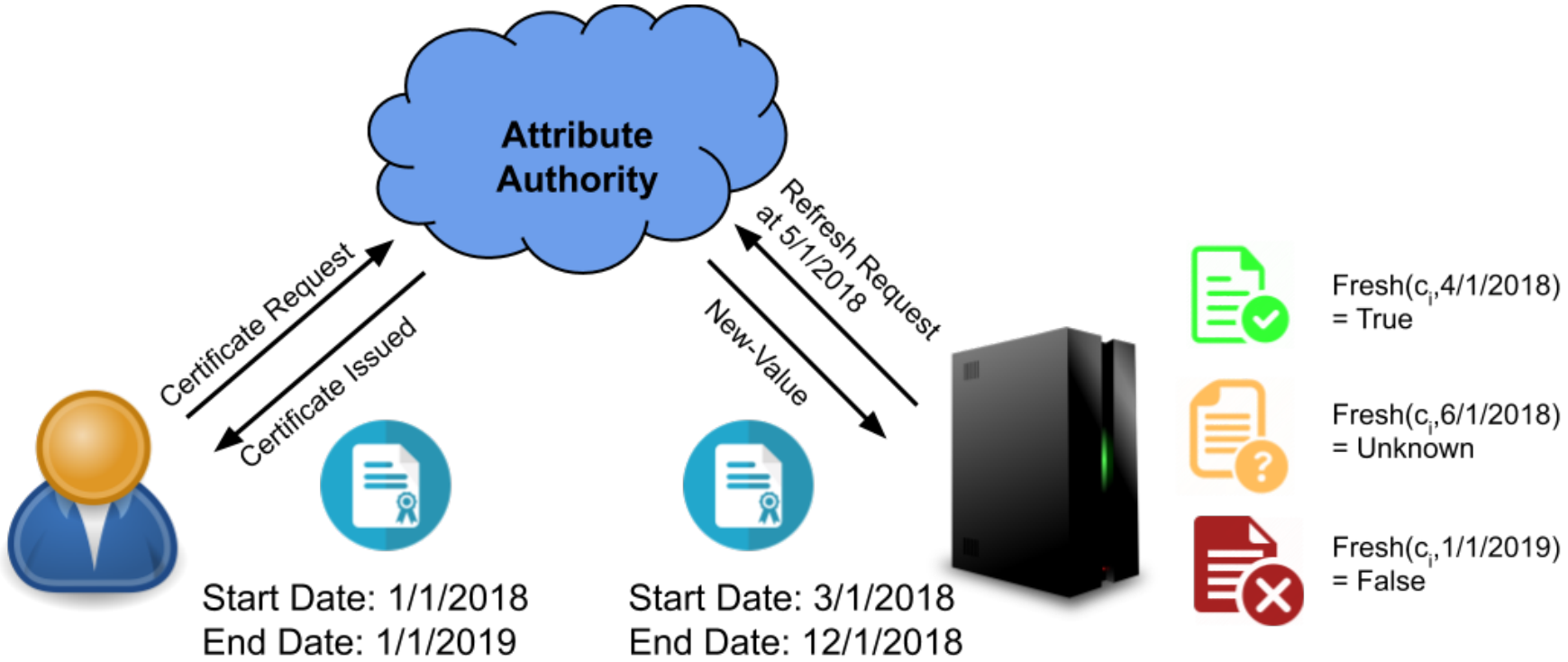
$$V_{DP}^{P,t} = \bigwedge_{1 \le i \le n} F(i), Sat_{c_i}^{P,t} = True \leftrightarrow F\left(val_{kmax(t)}^{i}\right) = True$$

✓For example, if the policy requires the security level to be at least 3 and requesting subject to be a manager, any security level credential with the value greater than or equal to 3 and any role certificate indicating the subject is a manager would be considered as satisfactory.

✓The same credential may be satisfactory with respect to one policy and not satisfactory to the other.

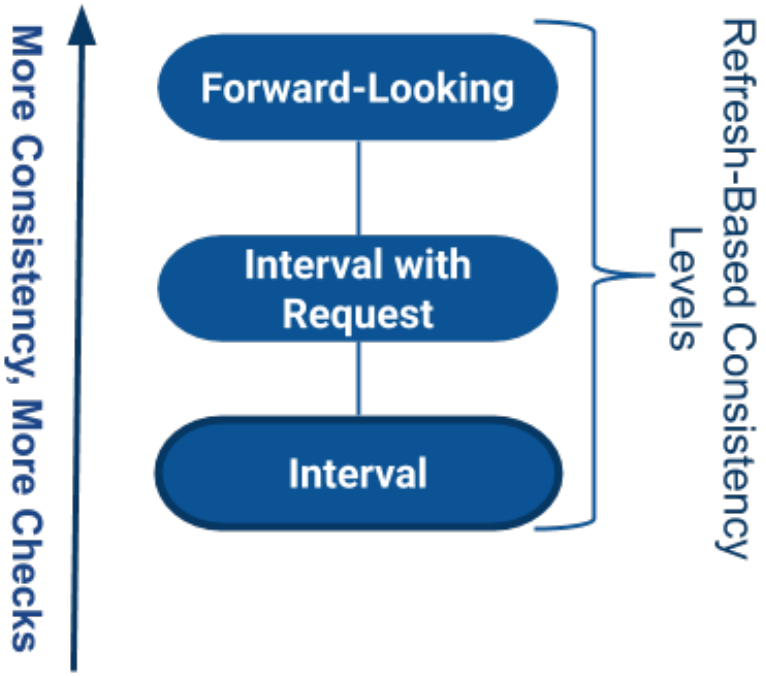
Preliminaries (cont.)

- Freshness: We assess an attribute value as *fresh*, when the current time falls in its lifetime before latest known refresh which has not returned *Invalid* result.



Consistency Levels

- Three increasingly powerful consistency levels, each imposing more restrictive constraints on timing and sequencing of attribute refresh checks, compared to the previous work on consistency in trust negotiation systems.



Example

- In a coding company, we have following policy to regulate access to project documents:

$$P = (role \in \{manager, testing\ engineer\}) \wedge (security\_level \geq 5)$$

Bob

Role Certificate Refresh Results:

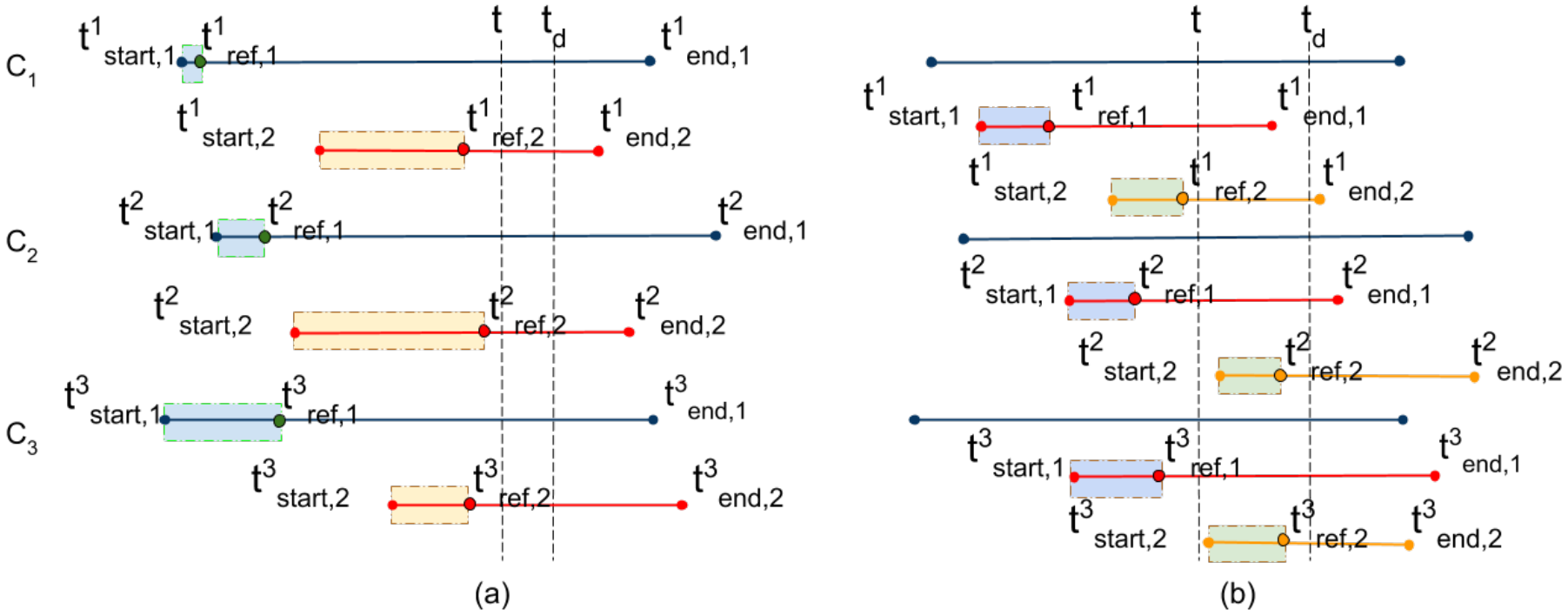
Refresh Date	Refresh Result	Role
Jan 15	Still-Good [Jan1,Jan 25]	Project Manager
Jan 21	New-Value [Jan 20, March 20]	Testing Engineer

Security-Level Certificate Refresh Results:

Refresh Date	Refresh Result	Security Level
Jan 15	Still-Good [Jan10, March 20]	6
Jan 28	New-Value [Jan 26, March 20]	4

Interval Consistency

- All relevant credentials found to be simultaneously fresh before the decision time, while having satisfactory values with respect to the policy.



Example

Role Certificate Refresh Results:

Refresh Date	Refresh Result	Role
Jan 15	Still-Good [Jan1,Jan 25]	Project Manager
Jan 21	New-Value [Jan 20, March 20]	Testing Engineer

Security-Level Certificate Refresh Results:

Refresh Date	Refresh Result	Security Level
Jan 15	Still-Good [Jan10, March 20]	6
Jan 28	New-Value [Jan 26, March 20]	4

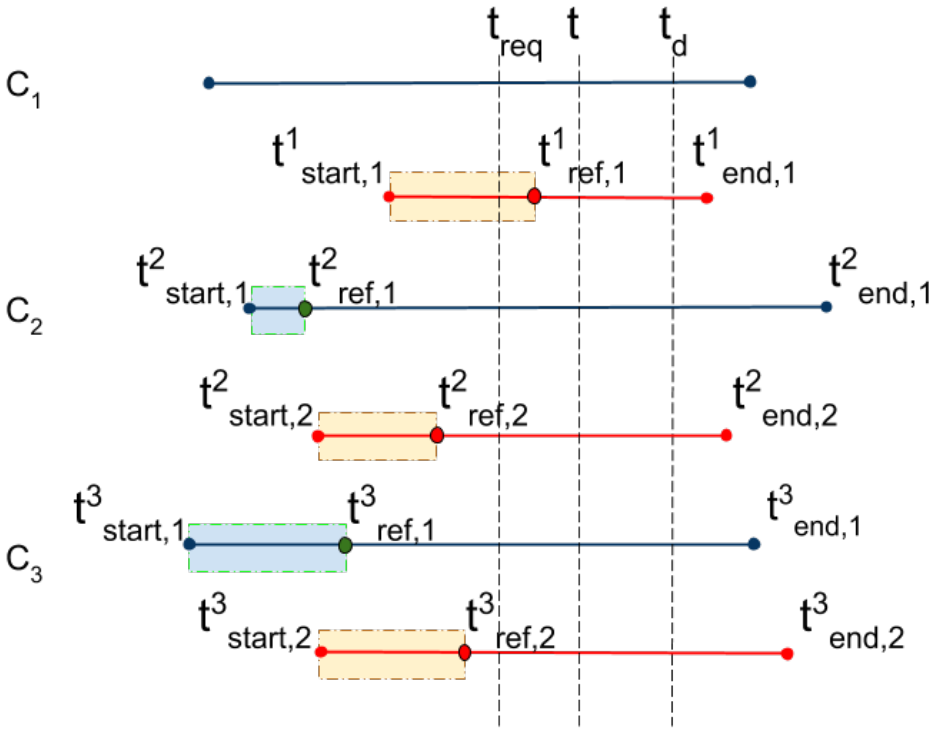
Access Request at Jan 25, 2:00 PM →

← Access Revoked in Revocation Scenario at Jan 25, 2:01 PM

← Access Granted in Refresh Scenario at Jan 25, 2:01 PM

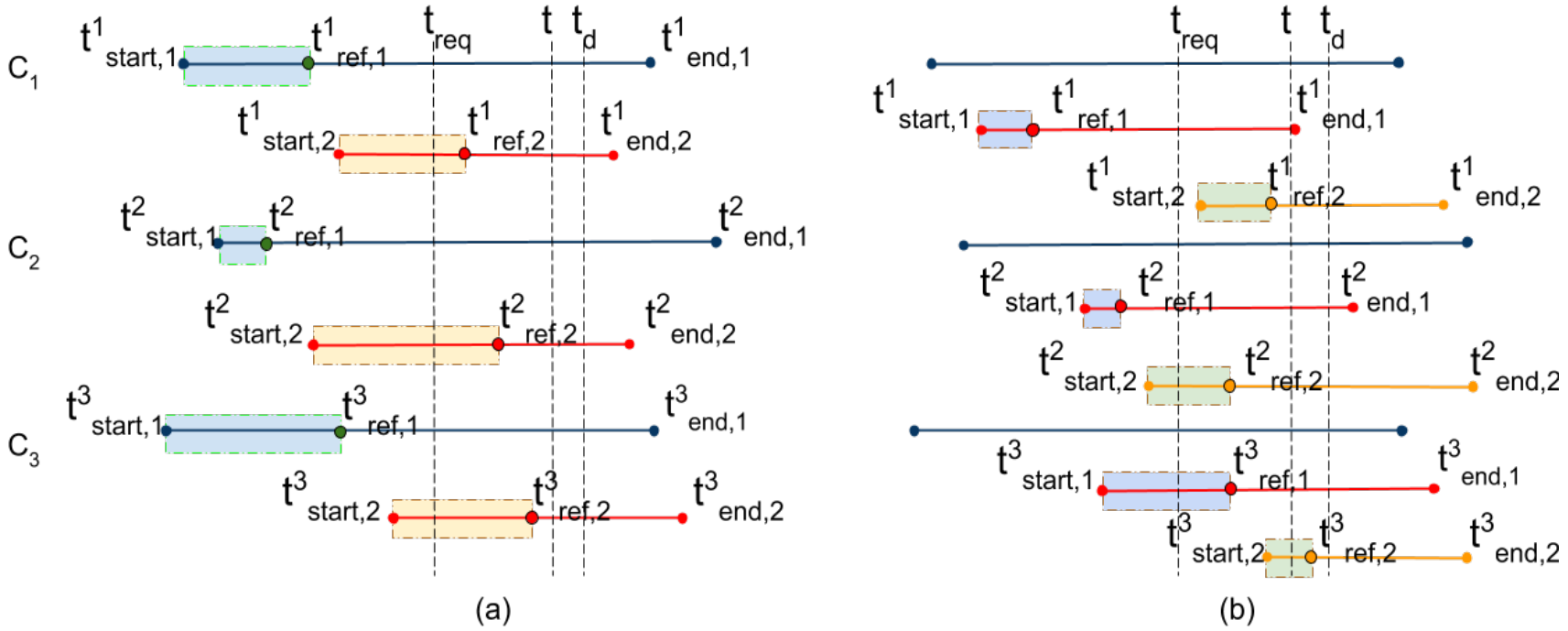
Interval Consistency with Request Time

- Unavailable refresh result for any relevant credential could be compensated after request time, before making any decision.



Forward-looking Consistency

- All credentials have been valid simultaneously at some point after the *request time*, considering both new and old values.



Example

Role Certificate Refresh Results:

Refresh Date	Refresh Result	Role
Jan 15	Still-Good [Jan1,Jan 25]	Project Manager
Jan 21	New-Value [Jan 20, March 20]	Testing Engineer

Security-Level Certificate Refresh Results:

Refresh Date	Refresh Result	Security Level
Jan 15	Still-Good [Jan10, March 20]	6
Jan 28	New-Value [Jan 26, March 20]	4

Access Request at Jan 20, 2:00 PM →

Access Revoked in Revocation Scenario at Jan 20, 2:01 PM ←

Access Granted in Refresh Scenario at Jan 20, 2:01 PM ←

Discussions

- There is a tradeoff between the availability and safety assurance provided by higher levels at the cost of additional checks.
- We have compared qualitative benefits provided by each level. However, quantifying cost-benefit is highly implementation and application specific.
- We formulate the authorization decisions and provide correctness and appropriateness of proposed criteria, however the underlying information could be attacked anytime during the process.

Conclusion and Future Work

- The safety and availability problem in multi-authority distributed ABAC systems has been formally characterized.
- The refresh scenario introduce instead of the traditional revocation check.
- We proposed three consistency levels which are totally ordered in strictness.

Some future research directions:

- Other access control information is subject to staleness, e.g. policy and object attributes.
- Attributes could be mutable.
- Models could be developed for ongoing authorization.

Thank You