# Safety and Consistency of Subject Attributes for Attribute-Based Pre-Authorization Systems

Mehrnoosh Shakarami(✉) and Ravi Sandhu

Department of Computer Science,
Center for Security and Privacy Enhanced Cloud Computing (C-SPECC),
Institute for Cyber Security (ICS), University of Texas at San Antonio,
San Antonio, USA
mehrnoosh.shakarami@my.utsa.edu, ravi.sandhu@utsa.edu

**Abstract.** Attribute-based access control (ABAC) systems typically enforce pre-authorization, whereby an access decision is made once prior to granting or denying access. This decision utilizes multiple components: subject's, object's and environment's attribute values as well as the authorization policy. Here, we assume that the policy, object and environment attribute values are known with high assurance while subject attributes are collected incrementally from multiple attribute authorities. This incremental assembly with differing validity periods for subject attribute values creates potential for inconsistency leading to incorrect access decisions. This problem was studied in context of trust negotiation systems by Lee and Winslett (LW), who define four different notions of consistency which are partially ordered in strictness. In this paper, we propose an alternate set of five consistency levels, also partially ordered in increasing strictness. Three of our levels are equivalent to counterparts in LW. The third LW level is differentiated by receive time, to which we are agnostic. Our fifth and highest level is new in that it utilizes request time which is not recognized in LW. We define the formal specification of each of our consistency levels and identify the properties guaranteed by each level. We discuss implication of these consistency levels in different practical scenarios and compare our work with related previous research.

**Keywords:** ABAC · Pre-Authorization · Safety · Consistency · Revocation

## 1 Introduction

In attribute-based access control (ABAC) access decisions are made on basis of attribute values of subjects, objects and environment with respect to a policy. For convenience we understand the term attributes to mean attribute values. Attributes and policy are susceptible to change. Ideally the decision point should know their real-time values, which is not practically feasible. Even if attributes

K.-K. R. Choo et al. (Eds.): NCS 2019, AISC 1055, pp. 248–263, 2020.
https://doi.org/10.1007/978-3-030-31239-8_19

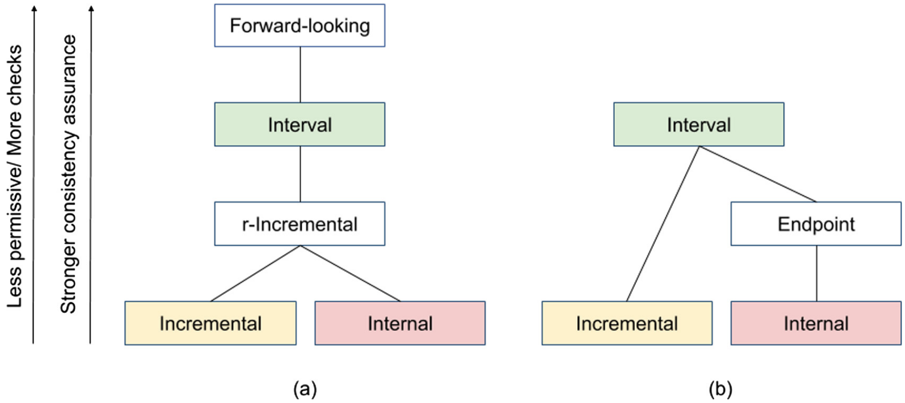

**Fig. 1.** (a) Our consistency levels (b) LW consistency levels. Equivalence is color coded.

and policy are queried from appropriate authorities immediately prior to every decision, there are irreducible network latencies. Realistically, some values will be cached due to performance, cost, and failures. Consequently, some access decision may be incorrect. We call this the safety and consistency problem.

This paper investigates this problem with focus on subject attributes. We assume that the policy and object/environment attributes are known in real-time at the decision point. This is reasonable since the decision and enforcement points are typically co-located with the object's custodian who maintains these values. This reduces the problem to safety and consistency of subject attributes. These attributes are obtained as credentials (a.k.a certificates) issued by an Attribute Authority (AA). Credentials may be signed or unsigned depending on how they are acquired. The closest prior work is by Lee and Winslett (LW) [8,9]. Our work is inspired by LW but takes a significantly different perspective and provides novel insights as discussed in Sect. 2.

This paper is organized as follows. In Sect. 2, we review the LW model and compare it to our work. Section 3 documents our system model and assumptions. In Sect. 4 we formalize proposed consistency levels specifications along with the properties guaranteed by each level. We review other relevant related works in Sect. 5. The implications of our consistency levels in context of different architectures and practical scenarios are discussed in Sect. 6. Section 7 gives our conclusion.

## 2 Lee-Winslett (LW) Model and Comparison

Figure 1 summarizes the relationship between the consistency levels defined in this paper (Fig. 1-(a)) versus the LW definitions (Fig. 1-(b)). A higher level requires additional checks, so it imposes a performance cost at the benefit of stronger consistency. Although formulated differently, three of the levels on both sides are equivalent as indicated by the same name and colors. Proofs of these equivalences are given in the appendix.

Common to both sets of definitions, a credential specifies the value of a single attribute for the subject. Each credential has a *start time* and *end time*, which establish the overall lifetime for validity of the attribute value given in the credential. The credential may be revoked by the AA during this putative lifetime. Thus the relying party (the decision point) must make one or more revocation checks with the AA to gain additional assurance of the credential's validity.[1] The consistency problem arises when multiple credentials of a subject are required to make an access decision. If the lifetimes and *revocation check times* of required credentials do not align properly it is possible to make incorrect access decisions, both in allowing access that should be disallowed and vice versa. This is discussed further and elaborated by examples in Sect. 3. The consistency levels of Fig. 1 define different criteria for aligning the lifetimes and revocation check times for multiple credentials.

The start, end and revocation check times are common to both sets of definitions.[2] The LW definitions all utilize the notion of *receive time*, which is the time when a credential is received by the relying party. Top two LW levels (endpoint and interval) utilize the notion of decision time. We consider decision time as central while receive time is irrelevant. We are agnostic about the delivery path and timing of credential receipt. The decision time is the instant where an access decision is committed and intuitively should be central to consistency consideration. Three of the LW levels are equivalently reformulated in our definition as indicated in Fig. 1. If receive time is ignored the endpoint and interval levels of LW are equivalent and hence not differentiated in Fig. 1-(a). The two new levels in our definitions, r-incremental (short for restricted incremental) and forward-looking are discussed below.

The incremental and internal levels allow the use of credentials that are known to be expired or revoked. So, we do not recommend general use of these two levels.[3] The explicit use of decision time enables us to formulate the r-incremental level that eliminates use of expired/revoked credentials. It is the lowest level we would generally recommend, and is missing in LW. The highest forward-looking level in our definitions requires the use of *request time*, which is the time when a subject makes a request for access and is missing in LW.

[1] Credential lifetimes can range widely from months to seconds. For very short-lived credentials revocation checks may not be useful. For simplicity we consider that for a short lived credential there is an implicit and successful revocation check at its start. Thus we can uniformly assume there is at least one revocation check by the relying party for each credential that it uses in making an access decision. For long-lived credentials there should be at least one revocation check after start time.

[2] Note that start and end times are determined by the AA, while revocation check times are determined by relying party actions.

[3] In a risk-based approach it may be acceptable to use expired/revoked credentials, but general use is not recommended.

## 3 Problem Statement and System Assumptions

The value of an attribute of a subject is represented by a credential which must be coupled to a specific subject, which is typically achieved by embedding the subject's identity in the credential. The identity of the subject must be authenticated before the credential is coupled with that subject. The details of these processes can be complex and susceptible to security vulnerabilities and flaws. All the same there are multiple well-known standards such as X.509 [1], SAML [11] and OAuth [15] in this arena. We assume that suitable mechanisms exist to bind credentials to subjects without requiring any specific technique for this purpose. Regardless of the way through which the attributes are presented, we assume proposed attributes to be authentic and tied to the subject.

We require every credential to have a determined lifetime interval which has been specified by its *start time* and *end time*. For short-lived credentials this interval is small, say minutes, seconds or even less, while for long-lived credentials this interval could span days, months or years. In either case we recognize the possibility that the credential may get revoked during its lifetime, although this is especially germane for long-lived credentials. We forbid use of a credential outside its lifetime. The revocation status of a credential may be checked as appropriate by the decision point, and a credential that is known to be revoked cannot be unrevoked. We also assume that attribute values do not change as a result of credential usage, so that attributes are immutable in the sense of [13].

Following LW we refer to the set of a subject's credentials used to make an access decision as the *view* of the decision point ($V_{DP}$). The appropriate view depends upon the policy being evaluated. In general, the view might change during evaluation. Consider a policy $P$ which is a disjunction of two predicates $A$ and $B$, i.e., $P = A \vee B$. The decision point may choose only credentials included in the $A$ predicate as relevant to $P$ in order to perform first step of evaluation; if $A$ fails, $B$ will replace it in the view and the set of relevant credentials to $P$ will change as a result.

**Definition 1.** *The set of attributes included in the view of decision point related to the policy $P$ at time $t$ is called the set of relevant credentials and denoted by $V_{DP}^{P,t}$.*

Since required credentials for evaluating a policy would be collected incrementally and lifetimes of different credentials might not be the same, there is no guarantee that previously collected credentials are still valid while the latter ones are acquired, which might cause the safety and consistency problem. Following example illustrates the inconsistency problem.

*Example 1.* Alice is a portal manager in the sales department of a company. If she wants to communicate with clients through the portlet website, the decision point needs credentials attesting her sales group membership and user role. If she wants to utilize higher levels of access, for example editing and approving contracts with clients, both user and manager role credentials are required. Suppose Alice has the user role since January 1st (start time) until March 1st (end time). Her

**Table 1.** Table of symbols

| Symbol | Meaning | Symbol | Meaning |
|---|---|---|---|
| $c_i$ | $i^{th}$ credential | $t_{req}$ | Request time |
| $t^i_{r,k}$ | Time of $k^{th}$ revocation check for $c_i$ | $t_d$ | Decision time |
| $t^i_{r,max}$ | Last time of revocation status check for $c_i$ | $t_e$ | Enforcement time |
| $t^i_{invalid}$ | First time $c_i$ has been found to be revoked | $t^i_{start}$ | Start time of $c_i$ |
| $t^i_{revoc}$ | Actual revocation time for $c_i$ (if any) | $t^i_{end}$ | End time of $c_i$ |

sales group member certificate is valid since January 25th till February 24th and she is given the manager role from Feb 10th (start time), which is valid until March 9th. Suppose the decision point acquired and validated the user role and sales group certificates most recently at January 25th and Feb 8th respectively. It also collected manager certificate at Feb 10th which was verified to be valid (via revocation check) on the same day. So, the decision point would honor the manager access to Alice if she request an access afterward. Due to a reorganization in the company, Alice may no longer be a manager after Feb 17th. Also, suppose Alice's user role certificate has been prematurely revoked at Feb 9th. But if the decision point still relies on the previous revocation checks, Alice would be able to exercise manager's rights after revocation of her relevant credentials, which results in an access violation.

Violation by relying on outdated validation information is a common problem in access control enforcement. While this example illustrates inconsistency with long-term cached credentials, similar problems can arise even if all needed credentials are accumulated over a short period of time.

Table 1 defines a set of self-explanatory symbols to refer to important time stamps used in this paper. There are some common assumptions which have been made in both LW and our work as follows.

1. Once a credential is revoked, it cannot be un-revoked. However, a new credential can be issued for the same attribute for that subject.
2. There is a single instantaneous decision time $t_d$. The access decision may be re-evaluated subsequently with, say, different credentials but this latter evaluation is treated as a separate and distinct decision.
3. $V^{P,t_d}_{DP}$ is the only view of interest, in which $P$ is the policy which should be satisfied to grant the access at decision time $t_d$ and $DP$ stands for the decision point.

We always use the latest revocation check results for making access decision. So, if we have $max_i$ revocation checks for $c_i$, the $t^i_{r,max}$ indicates the latest revocation check (${max_i}^{\text{th}}$) of $c_i$ and it would be utilized in decision making.

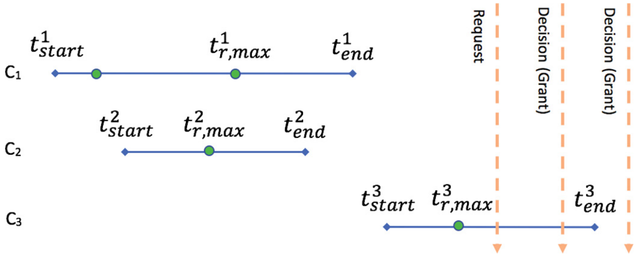

**Fig. 2.** Incremental consistency with unrestricted decision time

## 4 Consistency Levels

In this section, we develop five consistency levels relative to the view of the decision point ordered as shown in Fig. 1-(a). We say that a credential is in its validity interval or is *valid*, provided that the current time is not before the credential's start time, nor after the credential's end time and the credential is not known to be revoked at any time before the current time. A revocation check is never done after the end time since the credential has already expired. Once it is revoked a credential cannot become valid again. It follows that if validity of a particular certificate is confirmed via revocation check at time $t$ after its start time, the credential has been valid for all times between its start time and $t$.

Let $C$ be the set of all credentials in the system, and $T$ the set of all possible time stamps. We formalize the notion of a credential's validity status at time $t$ by defining the following 3-valued function. We call a credential *Invalid*, if following function returns *False*.

$$Valid : C \times T \rightarrow \{True, False, Unknown\}$$

$$Valid(c_i, t) = \begin{cases} True \iff & Valid(c_i, t^i_{r,k}) \wedge (t^i_{start} \leq t \leq t^i_{r,k}) \\ Unknown \iff & Valid(c_i, t^i_{r,max}) \wedge (t^i_{r,max} < t \leq t^i_{end}) \\ False \iff & (\neg Valid(c_i, t^i_{r,max}) \wedge (t \geq t^i_{r,max})) \\ & \vee (t \notin [t^i_{start}, t^i_{end}]) \end{cases} \quad (1)$$

In the rest of this section we propose five consistency levels. For each consistency level, we provide a formal specification along with the properties which are guaranteed if we apply the proposed specification.

### 4.1 Incremental Consistency

This level requires each relevant credential to be found valid by a revocation check before the decision time. In Example 1 suppose Alice wants to access the portal on Feb 25th, so user and sales group certificates should have been checked. Although one of her relevant credentials (sales group) has expired one day ago, the system would let her in. This access violation happens because in this level

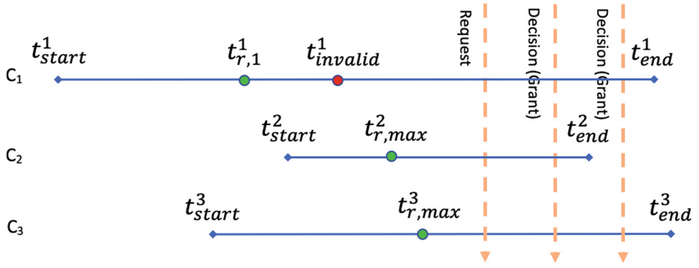

**Fig. 3.** Internal consistency

we may use a credential for an access decision after its $t^i_{end}$ time. As shown in Fig. 2, two credentials $C_1$ and $C_2$ have been used after their corresponding end times.

**Specification.** Every credential in the view of decision point is valid at its latest revocation check which has been done before the decision time.

$$(\forall c_i \in V_{DP}^{P,t_d})\ [(t^i_{start} \leq t^i_{r,max} < t^i_{end}) \wedge (\max_{\forall c_i \in V_{DP}^{P,t_d}} t^i_{r,max} < t_d) \wedge \mathit{Valid}(c_i, t^i_{r,max})] \tag{2}$$

*Property 1.* For every relevant credential, there is at least one point in time before the decision time, at which that credential has been found (via revocation check) to be valid.

$$(\forall c_i \in V_{DP}^{P,t_d})(\exists t_i)\ [(t^i_{start} \leq t_i < t^i_{end}) \wedge (t_i < t_d) \wedge \mathit{Valid}(c_i, t_i)]$$

*Proof.* Without loss of generality, we can assume $t_i = t^i_{r,max}$. Moreover, we know that $\max_{\forall c_j \in V_{DP}^{P,t_d}} t^j_{r,max} < t_d \implies t_i < t_d$.

### 4.2 Internal Consistency

In order to enforce lifetime overlap for all relevant credentials, internal consistency requires every relevant credential to be started before the end point of any other relevant credential. Furthermore, if a credential is revoked, this revocation should happen after all credentials have started. As shown in Fig. 3, it is possible to deliberately utilize an already revoked credential in this level. Moreover, it is still possible to use a credential beyond its end time, as in incremental consistency. In case of Example 1, Alice would be granted access to the portlet even if we know her user role credential has been revoked at Feb 9. The formal specification is as follows.

**Specification.** Every credential in the view of decision point has to be started before the minimum endpoint of all credentials and has to be valid at some point before the decision time. The minimum known revocation of any relevant credential occurs after all credentials have been started.

$$(\forall c_i \in V_{DP}^{P,t_d})(\exists t_{r,k}^i)\ [(t_{start}^i \leq t_{r,k}^i < t_{end}^i) \wedge \mathit{Valid}(c_i, t_{r,k}^i) \wedge (\max_{\forall c_i \in V_{DP}^{P,t_d}} t_{r,max}^i < t_d)$$
$$\wedge (\max_{\forall c_i \in V_{DP}^{P,t_d}} t_{start}^i < \min_{\forall c_i \in V_{DP}^{P,t_d}} t_{invalid}^i) \wedge (\max_{\forall c_i \in V_{DP}^{P,t_d}} t_{start}^i < \min_{\forall c_i \in V_{DP}^{P,t_d}} t_{end}^i)] \qquad (3)$$

*Property 1.* There is at least one point in time at which all relevant credentials are in their $[t_{start}, t_{end})$ time intervals and are not known to be *Invalid*.

$$(\exists t')(\forall c_i \in V_{DP}^{P,t_d})\ [(t_{start}^i \leq t' < t_{end}^i) \wedge (\mathit{Valid}(c_i, t') \neq \mathit{False})]$$

*Proof.* The last condition in Eq. 3 provides overlapping of lifetimes of all relevant credentials. Also, there is at least one revocation check for every credential at which it has been found to be valid. So, there is at least one point, namely $t'$, in intersection of lifetime intervals of all credentials at which every credential is either checked and found to be valid before $t'$ (its validation status is unknown at $t'$) or it has not been checked yet (so it is valid at $t'$). If there is any credential which has been found to be revoked, $t'$ should be picked from the interval: $t' \in [\max_{\forall c_i \in V_{DP}^{P,t_d}} t_{start}^i, \min_{\forall c_i \in V_{DP}^{P,t_d}} t_{invalid}^i)$.

*Property 2.* There is no subset relationship between incremental and internal consistency levels.

*Proof.* It is possible to have an incrementally consistent view in which there is no overlap between lifetime intervals of all relevant credentials. So, it would not be internally consistent. On the other hand, there may be an internally consistent view at which we recognize a credential at its latest revocation check to be prematurely revoked, so thereby not incrementally consistent.

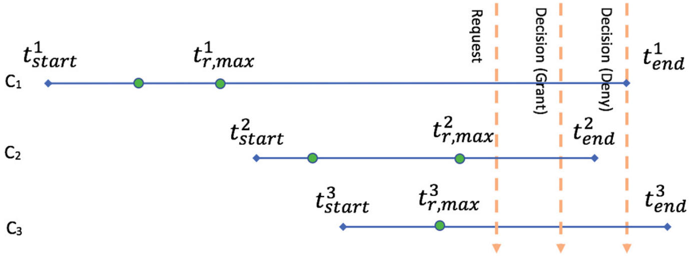

**Fig. 4.** Incremental consistency with restricted decision time

### 4.3 Incremental Consistency with Restricted Decision Time (Restricted-Incremental or r-Incremental)

In this level, we restrict the decision time to happen necessarily when all relevant credentials are in their lifetimes, say $[t_{start}^i, t_{end}^i]$ (Fig. 4). As opposed to previous levels, in this level if any of the relevant credentials has expired the access request would be denied. In case of Example 1, if Alice tries to exercise her rights at Feb 25th (after her credential expiration) the decision point would deny her access. In Fig. 4, the second decision time would result in Deny, comparing with the similar situation in Figs. 2 and 3 where both access requests resulted in Grant. Specification and guaranteed properties are given below.

**Specification.** Every relevant credential has to be found valid at the latest revocation check which, by assumption, happens before the decision time. Moreover, it is essential that the decision time happens before any of relevant credentials end time.

$$(\forall c_i \in V_{DP}^{P,t_d})\ [(t_{start}^i \leq t_{r,max}^i < t_d < t_{end}^i) \wedge \mathit{Valid}(c_i, t_{r,max}^i)] \tag{4}$$

*Property 1.* There is at least one point in time at which all the relevant credentials are in their $[t_{start}, t_{end})$ time intervals and are not known to be *Invalid*.

$$(\exists t')(\forall c_i \in V_{DP}^{P,t_d})\ [(t_{start}^i \leq t' < t_{end}^i) \wedge (\mathit{Valid}(c_i, t') \neq \mathit{False})]$$

*Proof.* Based on Eq. 4, $(\forall c_i \in V_{DP}^{P,t_d})\ [t_{start}^i \leq t_d < t_{end}^i]$. So, $\max_{\forall c_j \in V_{DP}^{P,t_d}} t_{start}^j \leq t_d < \min_{\forall c_j \in V_{DP}^{P,t_d}} t_{end}^j$ . By taking $t' = t_d$, the proof for the first part is trivial. For the second part, we know that the latest time we checked $c_i$'s revocation status is $t_{r,max}^i$, at which we found it to valid (otherwise the access would be denied). But, we do not know about the real status of the credential after the last revocation check and the *Valid* function would return *Unknown* at these later times.

*Property 2.* Any incrementally consistent view with restricted decision time has the following property: $\bigcap_{\forall c_i \in V_{DP}^{P,t_d}} [t_{start}^i, t_{end}^i) \neq \varnothing$

*Proof.* Following previous proof, there is at least one point $(t_d)$ that lies in the $[\max_{\forall c_j \in V_{DP}^{P,t_d}} t_{start}^j, \min_{\forall c_j \in V_{DP}^{P,t_d}} t_{end}^j)$ interval. So, this interval is not empty.

*Property 3.* Any r-incremental consistent view is incremental and internal consistent as well.

*Proof.* All three specifications have it in common that every relevant credential has to be found valid at its revocation check. The first part of the incremental consistency with restricted decision time is:

$$(\forall c_i \in V_{DP}^{P,t_d})\ [t_{start}^i \leq t_{r,max}^i < t_d < t_{end}^i \implies (t_{start}^i \leq t_{r,max}^i < t_{end}^i)$$
$$\wedge \ (\max_{\forall c_j \in V_{DP}^{P,t_d}} t_{r,max}^j < t_d) \wedge \exists t_d \in \bigcap_{\forall c_j \in V_{DP}^{P,t_d}} [t_{start}^i, t_{end}^i)]$$

Therefore, r-incremental is a constrained version (subset) of incremental level. Moreover, since we use the latest valid staus, we are not aware of any revocation and $t_{invalid}^i = \mathit{Null}$. So, all properties of internal level are also satisfied.

*Property 4.* It is not necessarily the case that any incrementally/internally consistent view is r-incremental as well.

*Proof.* In both incremental and internal levels, the decision time may be after some of the relevant credentials' endpoints, which means that we may have: $t_d > \min_{\forall c_i \in V_{DP}^{P,t_d}} t_{end}^i$ , which contradicts r-incremental specification.

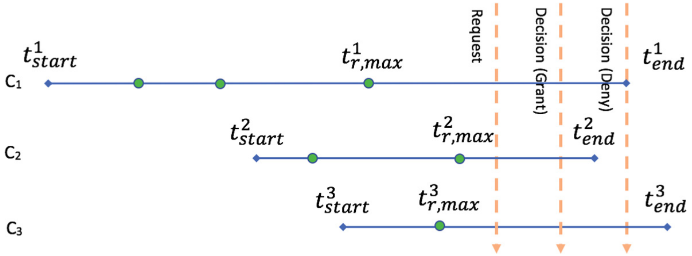

**Fig. 5.** Interval consistency

## 4.4 Interval Consistency

In Example 1, Alice's user role has been revoked at Feb 9th. If she tries to communicate at manager level with clients at that date and system still relies on the latest revocation check which happened before actual revocation she would be let in, while there is no guarantee that the credential is still valid. We know her user role has been revoked even before manager certificate starts. Interval level enforces latest revocation checks to happen in $[t^i_{start}, t^i_{end}]$ for all relevant credentials. So, it could be guaranteed that not only every credential is valid at some time, but also all credentials were simultaneously valid. The specification and properties guaranteed by this level are given below (Fig. 5).

**Specification.** Every relevant credential has found to be valid at the latest revocation check before the decision time. Moreover, the latest revocation check happened after all credentials have been started and before any of them ends.

$$(\forall c_i \in V_{DP}^{P,t_d})\ [(\max_{\forall c_i \in V_{DP}^{P,t_d}} t^i_{start} \leq t^i_{r,max} < t_d < \min_{\forall c_i \in V_{DP}^{P,t_d}} t^i_{end}) \wedge Valid(c_i, t^i_{r,max})] \tag{5}$$

*Property 1.* There is at least one point in time, after all relevant credentials have been started and before any of them ends, prior to decision time, at which all of the relevant credentials are simultaneously valid.

$$(\exists t')(\forall c_i \in V_{DP}^{P,t_d})\ [(\max_{\forall c_i \in V_{DP}^{P,t_d}} t^i_{start} \leq t^i_{r,max} < t' < \min_{\forall c_i \in V_{DP}^{P,t_d}} t^i_{end}) \wedge Valid(c_i, t')]$$

*Proof.* Let $t' = \min_{\forall c_i \in V_{DP}^{P,t_d}} t^i_{r,max}$. For every relevant credential to the policy, we could guarantee that it has been valid at $t'$. Note that if any credential has found to be revoked at $t'$, it cannot be unrevoked at any later time. Therefore, the proof is complete.

*Property 2.* Every interval consistent view is r-incremental.

*Proof.* It is trivial that: $(t^i_{start} \leq \max_{\forall c_i \in V_{DP}^{P,t_d}} t^i_{start}) \wedge (t^i_{end} \leq \min_{\forall c_i \in V_{DP}^{P,t_d}} t^i_{end})$. Substituting these equation in interval specification in Eq. 5, we can deduce: $(\forall c_i \in V_{DP}^{P,t_d})\ [t^i_{start} \leq t^i_{r,max} < t_d < t^i_{end}]$. So, interval specification satisfies the specifications of r-incremental.

*Property 3.* Not any r-incremental consistent view is necessarily interval consistent.

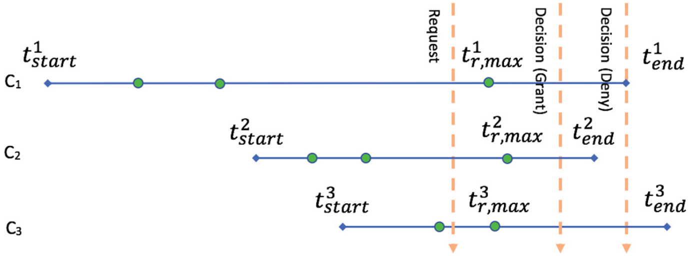

**Fig. 6.** Forward-looking consistency

*Proof.* Based on Eq. 4, latest revocation of a credential might happen before some credentials start time ($t^i_{r,max} < \max_{\forall c_j \in V_{DP}^{P,t_d}} t^j_{start}$) or a credential may be validated after some credentials expiration ($\min_{\forall c_j \in V_{DP}^{P,t_d}} t^j_{end} < t^i_{r,max}$), which contradicts with interval consistency specification.

### 4.5 Forward-Looking Consistency

In Example 1, suppose Alice tries to change the clients' contracts at Feb 17th (the set of relevant credentials includes sales group and manager role credentials). All relevant credentials were checked at Feb 10 at which all have been started and none of them expired yet, so the interval consistency timing constraints would be satisfied. However there is an access violation, because the decision point relied on outdated revocation status information (relying on revoked manager certificate). To solve this problem, we take the request time into account in our strongest level of consistency and impose constraints to ensure all credentials have been valid simultaneously at some point *after the request time* (Fig. 6).

**Specification.** Every relevant credential has to be valid at its latest revocation check time, which happens after the request time and before the decision time.

$$(\forall c_i \in V_{DP}^{P,t_d})[(\max_{\forall c_j \in V_{DP}^{P,t_d}} t^j_{start} \leq t_{req} < t^i_{r,max} < t_d < \min_{\forall c_j \in V_{DP}^{P,t_d}} t^j_{end}) \wedge \mathit{Valid}(c_i, t^i_{r,max})] \tag{6}$$

*Property 1.* There is at least one point in time, after the request time and before the decision time, at which all relevant credentials are valid simultaneously based upon their latest revocation checks.

$$(\exists t')(\forall c_i \in V_{DP}^{P,t_d})[(\max_{\forall c_i \in V_{DP}^{P,t_d}} t^i_{start} \leq t_{req} < t' < t_d < \min_{\forall c_i \in V_{DP}^{P,t_d}} t^i_{end}) \wedge \mathit{Valid}(c_i, t^i_{r,max})]$$

*Proof.* Suppose $t' = \min_{\forall c_i \in V_{DP}^{P,t_d}} t^i_{r,max}$. For every relevant credential, we could guarantee that it has been valid at $t'$, because otherwise it cannot be unrevoked at any later time including its latest revocation check. So, the proof is complete.

*Property 2.* Every forward-looking consistent view is interval consistent as well.

*Proof.* The definition of forward-looking consistency is a restricted version of interval consistency, in which we restricted the latest revocation check to happen necessarily after the request time.

*Property 3.* An interval consistent view is not necessarily a forward-looking consistent view as well.

*Proof.* In case of interval consistency, it is possible to have a credential $c_i$ with $t^i_{r,max} < t_{req}$, which contradicts with forward-looking consistency specification.

## 5 Related Work

Beyond the need to keep the data consistent in open and distributed systems, which has been discussed in the literature (see for example [17]), there is a crucial requirement to have access control models relying on the most recent information to grant/deny access to that data [3]. ABAC determines access based on attributes of subjects, objects and environment evaluated with respect to a policy. These attributes and the policy are exposed to change and staleness during the time which could result in inconsistency. On the other hand, delays and staleness of attributes are inherent in every distributed system owing to network latencies, caching and failures [5]. So, most practical distributed systems try to have a near-consistent [3] property, which attempts to limit the exposure of access control models to stale attributes.

The first organized work focused on consistency problem in trust negotiation proposed in [8,9]. Trust negotiation systems are specific types of distributed proof systems [6] which are appropriate when privacy is a major concern [14]. Lee and Winslett extend the proof construction in authorization systems to context-sensitive environments in [7], in which parts of the proof tree are required to remain hidden due to privacy concerns. In another research work on conversational web services [12], authors build their access control model based on user's credentials which relies on the first, most permissive level of consistency introduced in [8,9]. Authors simply assume the validity of a credential will last for the whole web service conversation duration.

Squicciarini et al. [16] present a protocol which safely performs trust negotiation during distinct negotiation sessions. Even though the authors put the probability of expired/revoked credentials during negotiation suspensions under consideration, they only mention that a synchronization algorithm would take care about updating the list of credentials without concretely describing the underlying synchronization scheme.

There is another category of research work which considers policy changes as the main concern. In [2], authors concentrated specifically on policy consistency in dynamic environments. In this paper, we consider policy inconsistency out of scope and assume that policy is known with high assurance at the decision point. Similarly, policy consistency is simply assumed as an underlying assumption in other research works [10].

Another closely related research to ours is [4], in which the authors formally specify a set of attributes in linear temporal logic in a Group-based Secure Information Sharing (g-SIS) model to express freshness of attributes. By proposing different levels of stale-safe property, authors try to limit unsafe access decisions relied on stale subjects' and objects' attributes in a distributed access control system.

Our approach differs from fail-secure access control models [18] since we assume that we can acquire fresh revocation status of credentials, but fail-secure access control applies in scenarios in which revocation states cannot be updated or accessed. Our main concern is credentials' revocation status might became obsolete since the last status update.

## 6 Discussion

**Implementation Implications.** Higher level of consistency requires additional checks. There is a tradeoff between the safety assurance provided by higher levels and cost of additional checks, as follows.

- From incremental/internal to r-incremental: the end times relate to decision time, so additional check of end time is required at the decision point in r-incremental.
- From r-incremental to interval: it potentially requires additional revocation checks, because all relevant credentials have to be checked at least once for their latest revocation status after all credentials have been started.
- From interval to forward-looking: all credentials definitely need to be checked for revocation status after the request time.

Quantitative performance evaluation is beyond the scope of this paper. It would require concrete system and workload assumptions and would be specific to the particular context.

**Short-Lived Credentials.** Short-lived credentials are used to obviate the need for revocation check by keeping credential lifetime very small. For our purpose we assume there is an implicit revocation check at start time, otherwise the AA would not issue the credential. No further revocation check is possible. In this case r-incremental and interval consistency will be equivalent. Forward-looking consistency could be guaranteed only if the request time has pushed prior to the start time for all credentials. The practical implication is that the decision point would need to assemble required subject credentials from appropriate AAs after the request time.

**Considering Enforcement Time.** After the decision point makes the access decision, it will be enforced by an enforcement point which could be the same or a different entity than the decision point. We certainly know that $t_d < t_e$. Proposed consistency levels in this paper remain unaffected by taking enforcement time into account. From another stand point, if there is a large gap between the decision and enforcement time, it is possible to utilize an access while some of the corresponding credentials have been expired; this is more probable in case of short-lived credentials. So, we can add more constraints to consistency level specifications which restricts this gap as follows: $t_e \leq \min_{\forall c_i \in V_{DP}^{P,t_d}} t^i_{end}$. Therefore, enforcement time could be considered to extend proposed levels of consistency.

## 7 Conclusion

Assuming an ABAC model is in place, we focused on a pre-authorization model in which our goal is to provide the decision point with the most recent status of subjects' attributes. To avoid safety and consistency problem, which is caused by relying on expired/revoked credentials, we proposed five increasingly powerful consistency specifications. At each level, we proved guaranteed properties provided by the proposed specification. We presented different implications of our proposed consistency levels in different real world scenarios and architectures. We also compared our work with the closest prior works and discussed its distinctive features and assumptions.

**Acknowledgement.** This work is partially supported by NSF CREST Grant HRD-1736209 and DoD ARL Grant W911NF-15-1-0518.

# A Appendix: Proof of Consistency Levels Equivalencies

We prove our claim of equivalent levels with LW model in this section. One of the distinctions is inequality of decision time with revocation check time, since we believe these two timestamps cannot be exactly the same, as the decision has to happen after revocation checks.

## A.1 Incremental Levels Equivalency

As seen in Sect. 4.1, for every relevant credential in our incremental level, there is at least one point before the decision time, at which that credential has been found to be valid. The incremental level in LW model is satisfied if and only if every credential to be valid at its receive time as follows.

$$\begin{aligned}(\forall c_i \in V_e^{P,t})\ [(\mathit{s.syn} = \mathit{True}) \wedge (\mathit{revocation\text{-}check}_i \neq \perp)\\ \wedge\ (\mathit{start}_i \leq \mathit{receive}_i \leq \mathit{revocation\text{-}check}_i)]\end{aligned}$$

This could be simplified as follows: $\mathit{start}_i \leq \mathit{receive}_i \leq \mathit{revocation\text{-}check}_i \leq \mathit{end}_i$. So, there is at least one point in time (receive time) at which every relevant credential has found to be valid, which matches with our incremental level. Moreover, this revocation check at the receive time could be considered as the latest validation. Then, we need to show revocation check in LW happens before the decision time, same as its counterpart in our model. Although the decision time has not been considered explicitly in LW model, revocation checks obviously happen before the decision time, since the receive time could not occur later than decision time. So, the proof is complete.

## A.2 Internal Levels Equivalency

Authors in LW define a view as internal consistent providing all relevant credentials satisfy the following conditions:

$$\begin{aligned}(\forall c_i \in V_e^{P,t})\ [\mathit{checked}(\mathit{credential\text{-}state}) \wedge (\max_{\forall c_j \in V} \mathit{start}_j < \min_{\forall c_i \in V} \mathit{invalidation}_i)\\ \wedge\ (\max_{\forall c_j \in V} \mathit{start}_j < \max_{\forall c_i \in V} \mathit{receive}_i) \wedge (\min_{\forall c_j \in V} \mathit{end}_j > \min_{\forall c_i \in V} \mathit{receive}_i)]\end{aligned}$$

Above conditions could be arranged as follows:

$$\begin{aligned}(\forall c_i \in V_e^{P,t})\ [(\mathit{start}_i < \mathit{revocation\text{-}check}_i \leq \mathit{end}_i) \wedge (\max_{\forall c_j \in V} \mathit{start}_j < \min_{\forall c_i \in V} \mathit{invalidation}_i)\\ \wedge\ (\max_{\forall c_j \in V} \mathit{start}_j < \max_{\forall c_i \in V} \mathit{receive}_i) \wedge (\min_{\forall c_j \in V} \mathit{end}_j > \min_{\forall c_i \in V} \mathit{receive}_i)]\end{aligned}$$

Based on our internal specification in Sect. 4.2, all conditions are the same except the last two conditions stated in LW model, which aim to provide an overlap between lifetime intervals of all relevant credentials in the view. Lifetime overlap has been provided in our model through $\max_{\forall c_i \in V_{DP}^{P,t_d}} \mathit{start}_i < \min_{\forall c_j \in V_{DP}^{P,t_d}} \mathit{end}_j$. Another distinction is the explicit consideration of decision time after all revocation checks. Even though this has not been stated in LW model, it is impossible to take revocation checks after the decision time into account while making decision, since it needs prediction of future states of credentials.

### A.3 Interval Levels Equivalency

To prove equality of the properties provided by both interval levels in our work and LW model, consider their definition of interval consistency for every relevant credential in the view:

$$(\forall c_i \in V_e^{P,t})\ [\mathit{checked}(\mathit{credential\text{-}state}) \\ \wedge (\mathit{start}_i \leq \mathit{receive}_i \leq \max_{\forall c_i \in V} \mathit{receive}_i \leq \mathit{revocation\text{-}check}_i)]$$

We can restate their interval definition as follows:

$$\mathit{start}_i \leq \mathit{receive}_i \leq \max_{\forall c_i \in V} \mathit{receive}_i \leq \mathit{revocation\text{-}check}_i \leq \mathit{decision\text{-}time} \leq \mathit{end}_i$$

The following property is concluded from above definition:

$$(\forall c_i \in V_e^{P,t_d})\ [\mathit{start}_i \leq \max_{\forall c_i \in V} \mathit{receive}_i \leq \mathit{revocation\text{-}check}_i] \\ \implies (\forall c_i \in V_e^{P,t_d})\ [\max_{\forall c_i \in V} \mathit{start}_i \leq \max_{\forall c_i \in V} \mathit{receive}_i \leq \mathit{revocation\text{-}check}_i]$$

On the other hand, we can formally deduce the following property from interval consistency definition in LW model:

$$(\forall c_i \in V_e^{P,t_d})\ [\mathit{revocation\text{-}check}_i \leq \mathit{decision\text{-}time} \leq \mathit{end}_i] \\ \implies (\forall c_i \in V_e^{P,t_d})\ [\mathit{revocation\text{-}check}_i \leq \mathit{decision\text{-}time} \leq \min_{\forall c_i \in V} \mathit{end}_i]$$

Putting above concluded properties together would result in the following definition. Taking out the receive time, this definition becomes the same as our interval definition.

$$\max_{\forall c_i \in V} \mathit{start}_i \leq \max_{\forall c_i \in V} \mathit{receive}_i \leq \mathit{revocation\text{-}check}_i \leq \mathit{decision\text{-}time} \leq \min_{\forall c_i \in V} \mathit{end}_i$$

## References

1. Housley, R., et al.: Internet X. 509 public key infrastructure certificate and CRL profile. Technical report (1998)
2. Iskander, M.K., et al.: Enforcing policy and data consistency of cloud transactions. In: ICDCSW. IEEE (2011)
3. Kortesniemi, Y., Sarela, M.: Survey of certificate usage in distributed access control. J. Comput. Secur. **44**, 16–32 (2014)
4. Krishnan, R., Niu, J., Sandhu, R., Winsborough, W.H.: Stale-safe security properties for group-based secure information sharing. In: FMSE. ACM (2008)
5. Krishnan, R., Sandhu, R.: Authorization policy specification and enforcement for group-centric secure information sharing. In: ICISS. Springer (2011)
6. Lee, A.J., Minami, K., Borisov, N.: Confidentiality-preserving distributed proofs of conjunctive queries. In: ASIACCS. ACM (2009)
7. Lee, A.J., Minami, K., Winslett, M.: Lightweight consistency enforcement schemes for distributed proofs with hidden subtrees. In: SACMAT. ACM (2007)
8. Lee, A.J., Winslett, M.: Safety and consistency in policy-based authorization systems. In: CCS. ACM (2006)

9. Lee, A.J., Winslett, M.: Enforcing safety and consistency constraints in policy-based authorization systems. In: TISSEC. ACM (2008)
10. Lee, A.J., Yu, T.: Towards quantitative analysis of proofs of authorization: applications, framework, and techniques. In: CSF. IEEE (2010)
11. OASIS: Security assertion markup language (SAML) v2.0 (2005)
12. Paci, F., et al.: ACConv–an access control model for conversational web services. In: TWEB. ACM (2011)
13. Park, J., Sandhu, R.: The UCON$_{ABC}$ usage control model. In: TISSEC. ACM (2004)
14. Peisert, S., et al.: Turtles all the way down: a clean-slate, ground-up, first-principles approach to secure systems. In: New Security Paradigms Workshop (2012)
15. RFC6749: The OAuth 2.0 authorization framework (2012)
16. Squicciarini, A.C., et al.: Identity-based long running negotiations. In: DIM. ACM (2008)
17. Steen, M.V., Tanenbaum, A.S.: Distributed Systems (2017)
18. Tsankov, P., et al.: Fail-secure access control. In: CCS. ACM (2014)