Access control for on-demand provisioned cloud infrastructure services
Ngo, C.T.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 6

Conclusion

In this chapter, we revisit questions in Chapter 1 and use research presented in Chapters 2, 3, 4 and 5 to answer them.

6.1 Answers to Research Questions

1. How do we design a flexible and scalable access control model supporting the on-demand provisioned self-service of cloud infrastructure services?

Chapter 2 answers this question. It introduced MT-ABAC, an ABAC model with multi-tenant properties integrating with the INDL cloud information model [79]. Inheriting the flexibility of ABAC, the proposed model can use attributes of subjects and resources in authorization policies. To support on-demand provisioned self-service, the model extended INDL by attaching policy templates binding with resource descriptions. Upon provisioning and rescaling, policies therefore can be generated and updated. Our prototype was implemented by an INDL extension for policy template descriptions. These templates were retrieved by SPARQL queries to compile into policies using a subset of the XACML language.

2. How does the proposed access control model adapt the virtualized resource sharing for multiple customers, who can manage their own resources not only in a single domain but also in multiple distributed domains?

Our proposed approach MT-ABAC in Chapter 2 extended the generic ABAC model to support multi-tenancy scenarios of the cloud systems. It distinguished general subjects into separated types, including providers, tenants and users. The capabilities and constraints of these subject types were defined to support the multi-tenancy features. This model therefore can manage virtualized resources pooling consumed by multiple customers at a cloud provider, where each of them could control authorization policies for their own subscribed resources.

To apply the MT-ABAC model for inter-provider scenarios, in Chapter 3 we provided token exchange approaches to solve distributed authorization and inter-domain security context management problems. In these cases, cloud resources can
interconnect between multiple providers. Our approach contained grant-tokens to relay decisions from the tenant’s domain to the provider’s domain via end-users. Upon validating grant-tokens, the system issued access-tokens for each provider’s domain, which then can be used by users to access resources. In this way, our approach allowed users to establish dynamic trust relationships with the chain of indirect providers for a specific authorization context.

3. How do we implement a high performance authorization policy evaluation engine, which should be required in access control solutions for cloud providers?

By applying XACML as the *de-facto* access control policy standard in our model, we provided the flexibility and expressiveness of the language in policy composition and management. However, to optimize XACML in high performance systems for cloud providers, our work in Chapter 4 analyzed the complex logic behind XACML components and evaluation processes. We then proposed the MIDD, a decision diagram-based mechanism with interval partition operations and MIDD operators. Our data structures and algorithms demonstrated in Chapter 5 to parse XACML policy tree into decision diagrams which substantially improve evaluation performance of the policy engine and the access control system. The proposed mechanism is reusable when it is able to apply not only for the policy evaluation, but also in various policy management problems.

6.2 Discussion

In this thesis, we identified challenges, models and mechanisms in building access control systems for cloud resource management at providers. Our solution specialized to use the ABAC model into multi-tenancy scenarios, with necessary constraints and capabilities for policy management. For practical implementations, we chose the *de-facto* attribute-based policy language XACML to apply in our model. Using XACML would make the approach having better compatibility and re-usability for different systems, not only in our project testbeds GEYSERS for cloud IaaS management, but also other cloud services systems.

The main drawback of XACML in applying for cloud system managements is performance issues for large-scale multi-tenant environments. Due to complex logic behind policies and algorithms, existing approaches cannot adapt high throughput requirements of cloud systems with thousands of customers. We solved this problem by analyzing XACML logic to propose data structures and algorithms for a high performance XACML policy evaluation engine. The implementation proved that our solution while still maintained policy semantics, can handle request throughputs with magnitudes faster than the standard implementation. Applied in the MT-ABAC for cloud infrastructure services of the DACI system, the testbed showed that the throughput can reach about 1400 requests/s for a scenario with 1000 customers, each having a separated, non-trivial virtual infrastructure. It is possible to not only scale-up the authorization systems with more robust computing powers, but also horizontally scale-out to increase overall system throughput because the policy evaluation mechanisms in our systems are stateless.
Our testbed was implemented and integrated successfully to the cloud infrastructure services in the GEYSERS project [10] for single and multiple providers scenarios. The design abstractions between cloud resource managements at the composition layers (a.k.a LICL layers in the GEYSERS project) and the virtual resource managements allow our approach to be able to integrate with different IaaS cloud stacks, not only the OpenNebula [84] in GEYSERS, but also others such as OpenStack or Eucalyptus. However, integrating with such stacks needs further real deployments and experiments. In other aspects, even though we tried to design MT-ABAC in the DACI system independent with cloud service models, our work contained primarily the testbed for IaaS model as in GEYSERS, while did not have validation work for PaaS and SaaS. Therefore, we have spaces to investigate and improve for other cloud services, which could be done in our future work.

6.3 Future Research

We propose two directions in the future research. The first line is to research on integrations and compatibilities of the MT-ABAC with other cloud service models and legacy systems. The second line is to analyze and compose access control policies using our techniques in Chapters 4 and 5.

Regarding integrating MT-ABAC to cloud systems providing other service models, resource description models managing cloud resources should be capable of on-demand provisioning and rescaling. Provider policies should be then generated and synchronized according to cloud provisioning and rescaling. Depending on desired scenarios between providers, tenants and end-users, related policies constraints should be defined and enforced at policy managements. In case of multi-domains scenarios, our token exchanges approach can be extended to establish, manage and validate trust and delegation management.

There are also open research on how to integrate and interoperate legacy access control systems at tenants' sides with the MT-ABAC. Authorization rules at legacy systems while can be composed by various policy languages should be validated and verified by defined constraints in MT-ABAC. The research should investigate on transforming legacy rules into the XACML, which facilitate for constraint verification and validation.

In other aspects, the MIDD data structures and algorithms used to solve the XACML evaluation performance problem can be applied to solve policy analysis and verification problems [100, 102]. Approaches in [100, 102] limited only to a subset of XACML language, while ours provide the most comprehensive policy semantic features. Other problems mentioned in section 4.6 such as policy testings, policy comparison, reverse authorization queries are also in our future researches on policy managements. Another section is to actively develop our open source XACML policy evaluation engine with enhanced features and profiles.

The development of cloud systems are becoming more popular. As a result, access control management for such systems should be aware to research further. Our work in this thesis regarding models, mechanisms and testbeds for access control are small but prospective contributions not only for Cloud Computing but
also for general access control research.