FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Diplomarbeitspräsentation

computer languages

Design and Implementation of a Security Model for the PeerSpace.NET

Masterstudium:
Informatikmanagement

Lukas Bitter

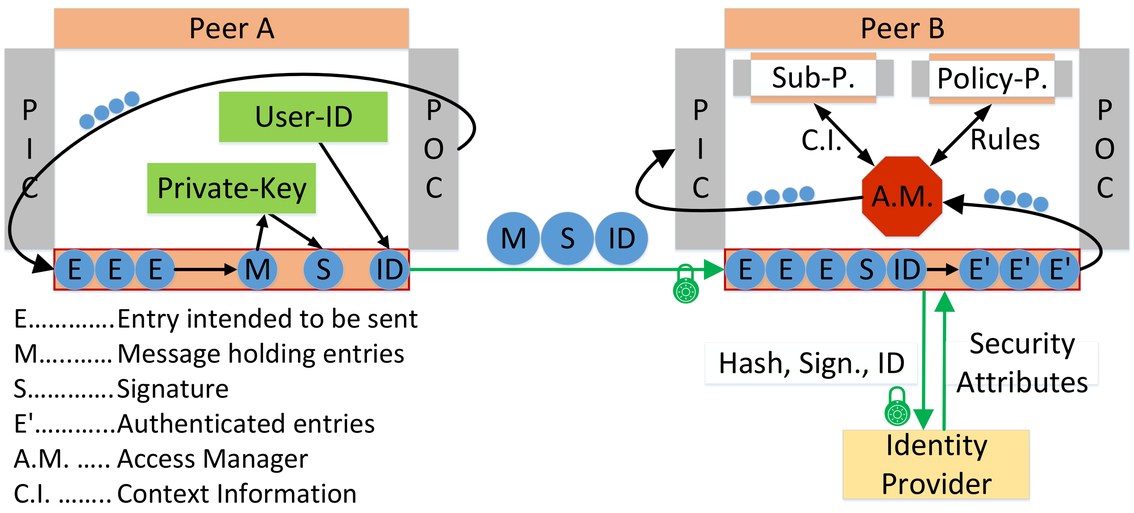
Technische Universität Wien
Institut für Computersprachen
Arbeitsbereich: Programmiersprachen und Übersetzer
BetreuerIn: A.o. Univ. Prof. Dr. Dipl.-Ing. eva Kühn

Goal and Research Question

- Design and implementation of a security model for the PeerSpace.NET
- PeerSpace.NET
  - Data-driven P2P-like coordination middleware
  - Services invoked via transmitted entries
- Security model
  - Fine-grained, attribute-based access control
  - Content- and context-aware rules
  - Involve all senders when entries are forwarded
- Research question
  - Is such a security model feasible for the PeerSpace.NET?
  - How usable is the security model?
  - How high are the performance losses?

The Peer Model and its Security Model

- Peer Model
  - Data-driven P2P-like coordination model
  - Composed of Entries, Peers, Containers, Wirings and Services
  - Containers store entries and form a peer's input/output stages
  - Wirings transport the entries and conduct the coordination logic
  - Services process the entries and conduct the application logic
- Security model rules write operations of entries from remote peers to containers



- Functioning of the security model
  - Sender signs entries and attaches the user id
  - Identity provider verifies the signature and provides the user's security attributes
  - Rules are obtained from the policy peer
  - Context information is obtained from arbitrary (sub-) peers
  - Receiver's access manager performs the access control
- An access control rule contains
  - Subject property template: identifies an entry's senders by their security attributes, supports delegation chains (e.g. when entries are forwarded)
  - Scope field: content information about the entry (e.g. type)
  - Condition: context information based on internal peer state, expressed via entries
  - Resource: peer and container
- Write operation is granted when at least one rule matches concerning subject property template, scope field, condition and resource
- Policy changes by writing rule entries to the policy peer
  - Possible from remote
  - Policy administration can be delegated

Motivation

Currently there exists no security mechanism for the PeerSpace-.NET, which is important for its practical employment. Designing and implementing a security model for the PeerSpace.NET, where no mutual trust can be assumed and no trusted server is available, is an interesting task.

Results and Evaluation

- Implementation of the security model was feasible
- Use case: academic exercise with student, tutor, supervisor and lecture server peers
  - Capability to express fine-grained rules (exceed access control lists), e.g: only *registered* students are allowed to upload *their own* solutions when the exercise is *enabled*
  - Straight forward creation and management of rules
  - Policy changes during run time are feasible
- Comparison of the security features to other models
  SPSN = implementation of the security model

	XVSM	Hermes	SMEPP	TuCSon	WCF	SPSN
RBAC	+	+	−	+	+	+
ABAC	+	−	−	−	+	+
Content-aware rules	+	+	−	+	∼	+
Context-aware rules	+	+	−	+	∼	+
Authorization for indirect sender	∼	−	−	≃	≃	+
Wildcard support for indirect sender	−	−	−	−	−	+
Dynamic policies	+	+	−	+	+	+
Remote policy changes	+	+	−	+	+	+
Administration delegation	+	+	−	+	+	+
Bootstrapped architecture	+	+	∼	+	−	+
Transparency	+	+	+	+	−	+
Scalability	∼	+	∼	∼	+	∼

- Benchmark tests
  - Time for putting entries into a peer
  - Different policies
    - Without access control (No AC)
    - With one/ten simple rules (SR): no scope field/condition
    - With one/ten complex rules (CR): with scope field/condition
  - Result: scales well for simple rules

Benchmark tests

Entries	No AC	1 SR	10 SR	1 CR	10 CR
100	78 ms	567 ms	580 ms	617 ms	701 ms
1000	168 ms	737 ms	743 ms	951 ms	1734 ms
10000	1569 ms	3148 ms	3252 ms	5165 ms	13033 ms

Conclusion

- Security model for the PeerSpace.NET has been implemented
- Attribute-based access control (ABAC)
- Content- and context-aware rules
- Supports delegation chains
- Dynamic policy changes
  - From remote
  - Policy administration can be delegated
- Good usability

Kontakt: bitter.lukas@gmail.com