Consumable Credentials in Logic-Based Access-Control Systems

Mohammad Hammoud
CS3525

Dept. of Computer Science
University of Pittsburgh
Introduction

• Prior work on access-control systems don’t support consumption of authority.

  Access-control frameworks permit a client to utilize the same credential for executing multiple jobs.

• Policies that use consumable authority that can be exercised only a limited number of times are needed (e.g., the authority to spend money).

• Linear logic (a type of logic in which an inference expends the premises that enabled it) can be used to model access-control systems.
  
  - Provides assurance that access control is implemented correctly (+).
  - Gives a clean and intuitive abstraction for reasoning about consumable credentials (+).
  - Credentials are transient (-).
The Proposed Work

1) They develop mechanisms that permit a decentralized logic-based authorization system to enforce the consumption of credentials (key contribution).

2) The consumable credentials are issued in such a manner that the credential’s use requires the consent of another entity, the ratifier.

3) The credential’s ratifier tracks the use of the credential and limits that use accordingly.

   - The moment of “use” at which the credential should be “consumed” is a subtle design with significant ramifications.

   - Not only must the credential’s consumption be enforced, but its availability must be protected against wasted consumption.

4) The key facets of the mechanism implementation are empirically evaluated.
Talk roadmap

- An illustrative access-control logic.
- Consumable credentials.
- Ratification:
  - Bounded use.
  - Atomicity.
- A working example.
- Quantitative evaluation.
- Concluding remarks (strengths and weaknesses).
Logic-Based Access Control

- **action**(*action, parameters, nonce*): denotes that it is OK to perform *action* during the session identified by *nonce*.
- **F true** means that there is exactly one copy of resource *F*.
- **F valid** means that *F* may be used any number of times.
- Hypothetical judgments: \( \Gamma; \Delta \vdash F \text{ true} \)

“by using each formula in delta exactly once and using the formulas in T any number of times, one can obtain one copy of *F*."

\[
\Gamma; F \text{ true} \vdash F \text{ true} \quad \text{(hyp)}
\]

\[
\Gamma, F \text{ valid}; \Delta, F \text{ true} \vdash G \text{ true} \quad \text{(copy)}
\]
Logic-Based Access Control

- **A affirms** \( F \): principal A affirms the truth formula of \( F \).

\[
\frac{\Gamma; \Delta \vdash F \text{ true}}{\Gamma; \Delta \vdash A \text{ affirms } F} \quad \text{(aff)}
\]

- To be able to write affirmations inside formulas, A affirms \( F \) is internalized as a connective **A says** \( F \).

\[
\frac{\Gamma; \Delta, F \text{ true} \vdash A \text{ affirms } G}{\Gamma; \Delta, (A \text{ says } F) \text{ true} \vdash A \text{ affirms } G} \quad \text{(saysL)}
\]

\[
\frac{\Gamma; \Delta \vdash A \text{ affirms } F}{\Gamma; \Delta \vdash (A \text{ says } F) \text{ true}} \quad \text{(saysR)}
\]
Logic-Based Access Control

- Direct affirmation: Alice signed $F$.

\[
\frac{\Gamma; \Delta, F \text{ true} \vdash A \text{ affirms } G}{\Gamma; \Delta, A \text{ signed } F \vdash A \text{ affirms } G} \text{(signed)}
\]

\[
\frac{\Gamma, A \text{ signed } F; \Delta, A \text{ signed } F \vdash G \text{ true}}{\Gamma, A \text{ signed } F; \Delta \vdash G \text{ true}} \text{(copy')}
\]

- Alice speaksfor Bob indicates that Bob has delegated to Alice his authority to make access-control decisions about any resource or action.

- delegate(Bob, Alice, action) transfers to Alice only the authority to perform the particular action called action.
Consuming Credentials

- How to utilize linear access-control logic in a distributed system implementation?

- In access-control context:
  - The hypotheses of a proof are credentials.
  - The proof shows that a policy (the proved formula) is satisfied by the credentials.

- The moment of “use” at which the credential should be “consumed” is a subtle design decision.
  - Consume a credential when a proof is verified by a reference monitor.
  - Consume a credential during proof construction when the linear inference rule (hyp) is used.

✓ Hypothesis consumption occurs as a step after the main search process for constructing a proof is completed, but before the proof is checked.
Ratification

- **Ratification** is an extra-logical step which is used to enforce the consumption of credentials, over and above the linear logic in which those credentials are expressed and used.

- Consumable credentials ($A \text{ signed}_k F$) are created with respect to a **ratifier** that monitors their use and enforces their consumption.

$$
\Gamma; \Delta, F \text{ true} \vdash A \text{ affirms } G \\
\Gamma; \Delta, A \text{ signed}_A, F \vdash A \text{ affirms } G^{(\text{signed}_L)}
$$

- Properties:

  1) **Bounded Use**: The system must enforce that the global number of uses of a consumable credential does not exceed that allowable uses as specified by the ratifier.

  2) **Atomicity**: Preventing capricious consumption of credentials during proof process. The process of ratification must either occur for all credentials, or none of them.
Ratification

- After the proof has been completed using consumable credentials, it must be sent to the applicable ratifiers who will certify that the consumable credentials are still valid.

- **Ratification credentials** are issued which the ratifiers sign.

\[
\Gamma; \Delta, F \text{ true} \vdash A \text{ affirms } G \\
\Gamma; \Delta, A \text{ signed}_A, F \vdash A \text{ affirms } G^{(\text{signed}_L)}
\]

- Ratification credentials cannot be made with respect to only the credential they are ratifying (to satisfy the **bounded use** property).

- The ratification credential is created additionally with respect to the proof in which it is included.

- A **contract-signing** protocol is run among the ratifiers for the consumable credentials used in the proof of \( F \) (to satisfy the **atomicity** property)
A Working Example

Alice walks into a store, fills her shopping cart with items and proceeds to check out. Instead of giving the clerk cash or credit card, she presents him with a proof that the store will be given its money.

• Bob, the store owner, is the reference monitor.

• Bob issues Alice a challenge describing the proof of payment that she must produce:

\[ G = \text{ACH says}\ action(pay,(Bob,\$100),\ nonce) \text{ true} \]

• Alice starts the proving process by stating her willingness to pay Bob:

\[ C_0 = \text{Alice signed}\ action(pay,(Bob,\$100),\ nonce) \]
A Working Example

• Alice must now demonstrate that there exists a chain of delegate and speaksfor relations from herself to the ACH.

\[
\begin{align*}
C_1 &= \text{BankA signed (Alice speaksfor BankA.Alice)} \\
C_2 &= \text{ACH.BC signed (BankA speaksfor ACH.BC.BankA)} \\
C_3 &= \text{ACH signed (delegate (ACH, ACH.BC, pay))} \\
C_4 &= \text{ACH.BC signed (delegate (ACH.BC, ACH.BC.BankA, pay))}
\end{align*}
\]

• Alice must now find a delegation statement allowing her to spend money from her account:

\[
C_5 = \text{BankA signed_{BankA} (delegate (BankA, BankA.Alice, pay))}
\]
A Working Example

• To obtain the ratification credential for $C_5$, Alice submits the proof to BankA's ratifier, RBankA.

• The ratifier deducts $100 from Alice’s account and transfers that money to the ACH.

\[ C_6 = \text{RBankA signed } \langle \text{delegate (BankA, BankA.Alice, pay), } M, \text{ACH says action(pay, (Bob, $100), nonce) } \rangle \]

• Alice now has a ratified proof which she submits to Bob for verification.

• Bob accepts the proof.
Quantitative Evaluation

The latency of the ratification protocol as a function of the number of ratifiers, includes the cost of credential verification, proof checking, and the creation, verification and communication of non-interactive zero-knowledge proofs.
Quantitative Evaluation

Breakdown of costs involved in the ratification protocol for each of the five ratifiers involved in a five-ratifier contract signing protocol.
Concluding Remarks: Strengths

• Permitting a decentralized logic-based authorization system to enforce the consumption of credentials.

• The system is flexible in that it permits the enforcement and straightforward specification of arbitrary, even dynamically determined limits on the use of credentials.

• Providing quantitative evaluation.

• Tight integration of their system to linear logic.
Concluding Remarks: Weaknesses

- The proposed contract-signing protocol is very expensive with respect to the required communication between ratifiers (each single ratifier participates with the other ratifiers to contribute its ratification credential). This degrades the latency of the ratifiers. (See Fig. 2). Running under n ratifiers, the protocol involves $O(n^3)$ messages in $O(n^2)$ rounds. Summarily, the cost of the contract-signing protocol dominates the proof-checking time.

- The system is not scalable: the latency of the ratification protocol highly increases as we increase the number of ratifiers (See Fig. 1).

- High storage overhead: a use counter is required per each credential.
Thank you!

Consumable Credentials in Logic-Based Access-Control Systems

M. Hammoud
Dept. of Computer Science
University of Pittsburgh