Chained and Delegable Authorization Tokens

G. Navarro  J. García  J. A. Ortega-Ruiz

Dept. of Computer Science
Universitat Autònoma de Barcelona

NordSec 2004
Outline

1. Introduction
2. Example
   - Initialization
   - Token delgation
   - Chain delegation
3. Delegation in CADAT
4. Implementation and Applications
   - Implementation
   - SPKI cert without using full tag intersection
   - SPKI cert using full tag intersection
5. Conclusions
1 Introduction

2 Example
   - Initialization
   - Token delgation
   - Chain delegation

3 Delegation in CADAT

4 Implementation and Applications
   - Implementation
   - SPKI cert without using full tag intersection
   - SPKI cert using full tag intersection

5 Conclusions
Chained And Delegable Authorization Tokens

- Hash chains as chains of authorization tokens.
  - tokens represent generic authorizations (not just micropayments).

- Delegation
  - delegation of chains or subchains.

- Implemented with a trust management infrastructure.
Chained And Delegable Authorization Tokens

- Hash chains as chains of authorization tokens.
  - tokens represent generic authorizations (not just micropayments).

- Delegation
  - delegation of chains or subchains.

- Implemented with a trust management infrastructure.
Introduction

Chained And Delegable Authorization Tokens

- Hash chains as chains of authorization tokens.
  - tokens represent generic authorizations (not just micropayments).

- Delegation
  - delegation of chains or subchains.

- Implemented with a trust management infrastructure.
Chained And Delegable Authorization Tokens

- Hash chains as chains of authorization tokens.
  - tokens represent generic authorizations (not just micropayments).

- Delegation
  - delegation of chains or subchains.

- Implemented with a trust management infrastructure.
Hash chains as chains of authorization tokens.
- tokens represent generic authorizations (not just micropayments).

Delegation
- delegation of chains or subchains.

Implemented with a trust management infrastructure.
Example: first use

AcmeNews

{ contract(acme,10) }

Alice

Generate hash chain:
h_{10}, h_{9}, ..., h_{1}
Example: first use

AcmeNews

{ contract(acme,10) }

Alice

Generate hash chain:
\[ h_{10}, h_9, ..., h_1 \]
Example: first use

AcmeNews

{ contract(acme,10) }

Alice

{ contract(h_10) }

Generate hash chain:

h_10, h_9, ..., h_1
Example: first use

AcmeNews

Alice

{ contract(acme,10) }

{ contract(h_10) }

h_9

Generate hash chain:

h_10, h_9, ..., h_1
Example: first use

Example

Initialization

AcmeNews

{ contract(acme,10) }

h_9

h_8

Alice

{ contract(h_10) }

Generate hash chain:

h_10, h_9, ..., h_1
Example: token delegation
Example: token delegation

\{ \text{token-deleg}(h_8) \}
Example: token delegation

{ token-deleg(h_8) }
Example: token delegation

{ token-deleg(h_8) }

h_7

h_6

ScienceNews
AcmeNews
Alice

G. Navarro et al. (UAB)
CADAT
NordSec 2004
Example: chain delegation
Example: chain delegation

\{ \text{chain-deleg}(h_6) \}
Example: chain delegation

AcmeNews  Alice  Bob

h_4

{ chain-deleg(h_6) }

h_5

h_4
**token-delegation**: delegatee is the consumer of tokens, who offers the service (aka *server-side* delegation).

**chain-delegation**: delegatee is the user of the tokens, who access the service (aka *client-side* delegation).
Delegation in CADAT

CADAT & Delegation

- **token-delegation**: delegatee is the consumer of tokens, who offers the service (aka *server-side* delegation).

- **chain-delegation**: delegatee is the user of the tokens, who access the service (aka *client-side* delegation).
CADAT is implemented in Java.

Contracts and delegations encoded as SPKI/SDSI authorization certificates.

Basic functionality provided by JDSI;
- Chain discovery algorithm → all computations needed by CATAD.
- Extended to support hash chain verification in the algorithm.
CADAT is implemented in Java.

Contracts and delegations encoded as **SPKI/SDSI authorization certificates**.

Basic functionality provided by **JSDSI**;
- Chain discovery algorithm $\Rightarrow$ all computations needed by CATAD.
- Extended to support hash chain verification in the algorithm.
CADAT is implemented in Java.

Contracts and delegations encoded as SPKI/SDSI authorization certificates.

Basic functionality provided by JSDSI;
- Chain discovery algorithm \(\Rightarrow\) all computations needed by CATAD.
- Extended to support hash chain verification in the algorithm.
Token as SPKI authorization certificate
Partial tag intersection

Authorization token: \( p = (cid, i, h^i(m)) \)

Token-cert without hash verification

```plaintext
(cert
  (issuer ...)
  (subject ...)
  (tag
    (h-chain-id |123456789|)
    (h-chain-index (* range numeric ge 7)))
  (comment
    (h-val
      (hash
        (md5 |899b786bf7dfad58aa3844f2489aa5bf|)))))
```
**Token as SPKI authorization certificate**

Partial tag intersection

Authorization token: \( p = (cid, i, h^i(m)) \)

**Token-cert without hash verification**

```
(cert
 (issuer ...)
 (subject ...)
 (tag
   (h-chain-id |123456789|)
   (h-chain-index (* range numeric ge 7)))
 (comment
   (h-val
    (hash
     (md5 |899b786bf7dfad58aa3844f2489aa5bf|))))
```
Token as SPKI authorization certificate

Full tag intersection

Authorization token: \( p = (cid, i, h^i(m)) \)

Token-cert with hash verification

(cert
  (issuer ...)
  (subject ...)
  (tag
    (hash-auth
      (hchain-id |lksjfSDFIstdMkJ0sndKISHfoMSKJSD|)
      (hchain-index 15)
      (hash md5 |d52885e0c4bc097f6ba3b4622e147c30|))))
Token as SPKI authorization certificate

Full tag intersection

Authorization token: \( p = (cid, i, h^i(m)) \)

Token-cert with hash verification

\[
\text{(cert} \\
\text{(issuer ...)} \\
\text{(subject ...)} \\
\text{(tag} \\
\text{ (hash-auth} \\
\text{ (hchain-id |lksjfSDFIsdkj0sndKISHfoMSKJS|}) \\
\text{ (hchain-index 15)} \\
\text{ (hash md5 |d52885e0c4bc097f6ba3b4622e147c30|)))}
\]

G. Navarro et al. (UAB)

CADA

NordSec 2004

11 / 15
Applications

- Generic token-based access control system.
- Micropayment schemes.
- Current application:
  - Token-based access control for mobile agents.
Applications

- Generic token-based access control system.
- Micropayment schemes.
- Current application:
  - Token-based access control for mobile agents.
Applications

- Generic token-based access control system.
- Micropayment schemes.
- Current application:
  - Token-based access control for mobile agents.
CADAT and mobile agent access control

- Alice
  - initial chain-contrac-cert
  - publish access tokens for her agents
  - Alice's mobile agent
  - token-contract-cert
- token Authority
  - verify contract, and accept tokens for M, if all verifications are successful
- Agent Platform 1
  - token-contract-cert
Conclusions

A system for token-based access control and micropayment systems.
- hash chains,
- delegation.

Implemented with SPKI/SDSI.

Current application: access control in mobile agent systems.
Conclusions

- A system for token-based access control and micropayment systems.
  - hash chains,
  - delegation.

- Implemented with SPKI/SDSI.

- Current application: access control in mobile agent systems.
A system for token-based access control and micropayment systems.
- hash chains,
- delegation.

Implemented with SPKI/SDSI.

Current application: access control in mobile agent systems.
Thank you! questions?