Security Framework for Decentralized Shared Calendars

Jagdish Prasad Achara

Research Master of Computer Science (Specialty : Services, Security and Networks)

24 juin 2011

Nancy-Université
Outline

1. Introduction
2. Security Requirements of DeSCal
3. State of the art
4. Proposed Security Framework
5. Implementation on iPhone OS
6. Possible Directions of Future Work
Outline

1 Introduction
   - Decentralized & third party independent shared calendar
   - About DeSCal
   - Problem Statement & Motivations
   - Challenges & Contributions

2 Security Requirements of DeSCal

3 State of the art

4 Proposed Security Framework
   - Security Framework Design Requirements
   - Security Framework Description
   - An illustrating example
   - Securing the communication between users
   - Discussion

5 Implementation on iPhone OS

6 Possible Directions of Future Work
Decentralized & third party independent shared calendar

- Shared Calendar?
- Why decentralized & third party independent?
  - Support for Ad-Hoc networks (802.11 networks).
  - No single point of failure.
  - Secrecy/confidentiality of shared calendar events.
  - Availability of data.

3rd party has all shared calendar information

Confidentiality?
Availability?
Considering the usefulness of such a decentralized shared calendar, DeSCal (abbreviation of Decentralized Shared Calendar) is proposed by us.

What is DeSCal?

An administrator of an event and his role?

A user can take two types of operation in DeSCal:

2. Administrative operation: On his access control policy to allow/deny other users to ‘Read’, ‘Delete’ & ‘Edit’ his events.

The design of DeSCal consists of four modules:

1. Coordination module: needs cooperative log
2. Access Control module: needs administrative log or admin log in short
3. P2P/Ad-Hoc Network
4. User Interface

DeSCal replicates whole shared calendar state (Shared Calendar, Cooperative log, Policies, Admin logs) for fault tolerance, availability and crash recovery.
Problem Statement & Motivations

- **Motivations**
  - Providing confidentiality to replicated shared calendar events.
  - Securing the communication between users.
Challenges & Contributions

► Challenges
  ► DeSCal’s characteristic features?
  ► Decentralized ‘Read’ access control?
  ► Dynamic group of users..

► Contributions
  ► Proposed a required security framework.
  ► Its implementation on iPhone OS.
Outline

1 Introduction
   - Decentralized & third party independent shared calendar
   - About DeSCal
   - Problem Statement & Motivations
   - Challenges & Contributions

2 Security Requirements of DeSCal

3 State of the art

4 Proposed Security Framework
   - Security Framework Design Requirements
   - Security Framework Description
   - An illustrating example
   - Securing the communication between users
   - Discussion

5 Implementation on iPhone OS

6 Possible Directions of Future Work
Security Requirements of DeSCal

- Providing confidentiality to replicated shared calendar events.
  - In Shared calendar, cooperative log, policy and admin log?

- Securing the communication between users.
  - Group communication?
Outline

1 Introduction
   - Decentralized & third party independent shared calendar
   - About DeSCal
   - Problem Statement & Motivations
   - Challenges & Contributions

2 Security Requirements of DeSCal

3 State of the art

4 Proposed Security Framework
   - Security Framework Design Requirements
   - Security Framework Description
   - An illustrating example
   -Securing the communication between users
   -Discussion

5 Implementation on iPhone OS

6 Possible Directions of Future Work
State of the art

With the absence of central authority, security of 1) replicated data & 2) messages exchanged between peers, is a challenging task.

▶ Overview
  ▶ Other decentralized shared calendars and collaborative environments.
  ▶ Securing replicated data.
  ▶ Secrecy by splitting.
1 Introduction
   - Decentralized & third party independent shared calendar
   - About DeSCal
   - Problem Statement & Motivations
   - Challenges & Contributions

2 Security Requirements of DeSCal

3 State of the art

4 Proposed Security Framework
   - Security Framework Design Requirements
   - Security Framework Description
   - An illustrating example
   - Securing the communication between users
   - Discussion

5 Implementation on iPhone OS

6 Possible Directions of Future Work
Security Framework Design Requirements

- DeSCal’s characteristic features e.g., fault tolerance, availability, crash recovery, dynamic access control must not be lost.

- On top of coordination and access control models already employed by DeSCal.

- Must preserve broadcast group communication of DeSCal.
Security Framework Description

It uses public key cryptography where authentication of public key is compulsory.

- Pathak & Iftode’s protocol
Security Framework Description

- Encryption Notations used:
  - Symmetric: $E_{K_e}(e)$ and $D_{K_e}(e)$
  - Asymmetric: $\{m\}_{K_u}$ and $\{m\}_{K_u^{-1}}$

- Description based on all possible happenings:
  - User-generated happenings
    1. Inserting a new event
    2. Deleting an existing event
    3. Editing an existing event
    4. Grant Read right
    5. Revoke Read right
    6. Grant/Revoke Delete/Edit right (Not Relevant)
  - System-wide happenings
    1. A new user joins the shared calendar group.
    2. An existing user leaves the group.
    3. A user goes off-line and then, comes on-line again.

- How fault tolerance is achieved in DeSCal?
- Surviving a crash.
- How availability of data is ensured?
Inserting a new event

\[ e' = E_{K_e}(e), \{ K_e \} K_{Owner}, \{ K_e \} K_{AuthUser_1}, \{ K_e \} K_{AuthUser_2}, \ldots \]

Granting ‘Read’ right

\[ i = \{ K_e \} K_{u_1}, \{ K_e \} K_{u_2} \]
Security Framework Description

- Concurrency Issues
  - ‘Read’ right revocation and ‘Edit’ concurrent operations

Initially, authorized to read and edit event 'e'

- Initially, authorized to read and edit event 'e'
- Edits event 'e' to 'f'

\[ E_{K_f}(f), \{k_f\}_{K_{u_3}}, \{k_f\}_{K_{u_1}}, \{k_f\}_{K_{u_2}} \]

- Owner of event 'e'
- Revoke read right to \( u_2 \) for event 'e'

- Concurrent Operations

???
Concurrency Issues

- ‘Read’ right grant and ‘Edit’ concurrent operations

Initially, authorized to read and edit event ‘e’

Initially, authorized to read event ‘e’

Initially, not authorized to read event ‘e’

Owner of event ‘e’

Edits event ‘e’ to ‘f’ (New key (k_f) generation because of immediate revocation)

Concurrent Operations

Will not be possible to decrypt as the symmetric key is changed in previous edition of this event by u_1

Right attribution for event ‘e’ to u_3

-Revoke read right to u_2 for event ‘e’

Initially, authorized to read event ‘e’

Right attribution for event ‘e’ to u_3

Owner of event ‘e’
An illustrating example

Inside the group G

U2 joins the group G again and retrieves all his shared calendar state in persistent storage.

New user u4 arrives and joins the group G. u4 receives the whole shared calendar state in persistent storage.

encrypted copy of shared calendar stored in persistent storage
Securing the communication between users and Discussion

- Securing the communication

\[ m' = \{ m, \text{counter} \} \]
\[ m'' = \{ m', \text{sig} \} \text{ where } \text{sig} = \{ \text{hash}(m') \}_{K_{ui}^{-1}} \]

- Discussion
Outline

1 Introduction
   - Decentralized & third party independent shared calendar
   - About DeSCal
   - Problem Statement & Motivations
   - Challenges & Contributions

2 Security Requirements of DeSCal

3 State of the art

4 Proposed Security Framework
   - Security Framework Design Requirements
   - Security Framework Description
   - An illustrating example
   - Securing the communication between users
   - Discussion

5 Implementation on iPhone OS

6 Possible Directions of Future Work
Implementation on iPhone OS

- RSA algorithm for asymmetric encryption and public/private key pair of size 1024 bits.
- For symmetric encryption, AES-128.

**Figure:** Calendar, Event Detail, Policy and Available Peers view
**Figure:** Selection of various attributes to insert a new rule in policy
Outline

1 Introduction
   - Decentralized & third party independent shared calendar
   - About DeSCal
   - Problem Statement & Motivations
   - Challenges & Contributions

2 Security Requirements of DeSCal

3 State of the art

4 Proposed Security Framework
   - Security Framework Design Requirements
   - Security Framework Description
   - An illustrating example
   - Securing the communication between users
   - Discussion

5 Implementation on iPhone OS

6 Possible Directions of Future Work
Possible Directions of Future Work

- Verification and Analysis of security framework.
- Standardize the communication protocol.
- Policy for users to join the shared calendar group.
- Some works (CP–ABE, Broadcast Encryption) to be explored if they can be used to satisfy security requirements of DeSCal while preserving its characteristic features.