Interactive Private Multi-Party Calendar Scheduling

Master Thesis

RWTH Aachen, Informatik 5
Lehrstuhl Prof. Decker
Motivation

- Popularity of calendar scheduling applications
- Appointments can be arranged easily

→ Increased productivity for companies and individuals
Appointment Scheduling Software Market: 9.80% CAGR to 2021 Says a New Research Report at ReportsnReports

PUNE, India, August 1, 2017 /PRNewswire/ --

Appointment scheduling software market analyst says one trend in the market is emergence of mobile appointment scheduling software solutions. The increasing penetration of mobile service providers fuels the demand for mobile phones. Mobile communication services have increased significantly in the rural areas. Low-cost cellular phones are popular in these areas, and enterprises adopt mobile communication services to increase profits.

The analysts forecast global appointment scheduling software market to grow at a CAGR of 9.80% during the period 2017-2021. The study was conducted using an objective combination of primary and secondary information including inputs from key participants in the industry. The report contains a comprehensive market and vendor landscape in addition to a SWOT analysis of the key vendors.
Motivation

What about privacy?

Central server: The scheduling service company learns about your timetable

Public inputs: Other participants can see your timetable
Motivation

Why should one care?

- Curious colleagues
- Social engineering attacks
- Digital footprint
- ...
Motivation

Can we do better?
Goal: Development of a Secure Calendar Application

- **Privacy**: Allows to agree on an appointment, but keeps timetables private
- **Scalability**: Applicable for real-world problems with up to 30 participants
- **Interactivity**: Scheduling protocol finishes in seconds or few minutes
- **Usability**: Exposes a clean graphical user interface
- **Workflow integration**: Integrates into established calendar applications
  - Microsoft Outlook
  - Thunderbird Lightning
  - Apple Calendar
- **Flexibility**: Scheduling schemes can be easily customized
Goal: Development of a Secure Calendar Application

A public discussion of new cryptographic techniques that allow to maintain privacy in data science.

Scheduling completed. The selected date is

Friday, March 2, 2018 from 14:00 to 15:00

Export appointment
Outline

- Preliminaries
- Use Case
- Conceptual Approach
- Implementation
- Live Demo
- Evaluation
- Conclusion
Secure Multi-Party Computation (MPC)

- Research field in cryptography since 1980s
- Practical only in the last couple of years

Evaluate arbitrary function \( f \) with a number of parties such that:
- Every party holds the correct function result at the end of the protocol execution
- No party is able to learn inputs of any other party

\[ f(x, y) \]

\( \rightarrow \) Peer-to-peer system that simulates a trusted third party
Secure Multi-Party Computation (MPC)

Protocols:
- **GMW**: $n$-party protocol based on secret sharing
- **Yao**: asymmetric two-party protocol based on garbled circuits

Function representation:
- Boolean circuit (XOR and AND gates)
- Arithmetic circuit (Addition and multiplication of integers)

→ Peer-to-peer system that simulates a trusted third party
iCalendar Standard (RFC 5545)

- Text-based file format for calendar data
- Supported by most calendar software for importing and exporting calendars
- Commonly used for sending proposed dates via email
- Includes events, free and busy time slots, ...

BEGIN:VCALENDAR
VERSION:2.0
BEGIN:VEVENT
SUMMARY:PETs4DS Working Group Meeting
ATTENDEE;CN=Alice;PARTSTAT=ACCEPTED;EMAIL=alice@rwth-aachen.de
ATTENDEE;CN=Bob;PARTSTAT=NEEDS-ACTION;EMAIL=bob@rwth-aachen.de
DTSTART;TZID=Europe/Berlin:20170812T143000
DTEND;TZID=Europe/Berlin:20170812T153000
END:VEVENT
END:VCALENDAR
Use Case
Use Case

Focus: Employees within a company or a working group

Workflow:

1. Initiator publishes invitation (description of event, time slot candidates, …)
2. Recipients willing to attend the appointment connect to initiator
3. Every participant selects personal free and busy time slots
4. A secure computation protocol is executed on private inputs
5. Result is the scheduled date
Network Assumptions

- Direct communication by IP address/port number (no NAT, firewall, …)
- Network protected against external access
Potential Threats

• **Semi-honest adversaries** who try to reconstruct private information of other participants based on exchanged messages/protocol state

• **Malicious adversaries** who actively modify or drop messages in order to learn private data

• **Non-participating adversaries** who eavesdrop on the communication from within the network
Potential Threats

Consider only semi-honest adversaries

• Software restriction policies that prevent custom code from being executed
• Mutual trust in companies to a certain extent
• Shared goal of “getting the job done”
• High risk, low reward
  – Disciplining measures/poisoned working atmosphere when detected

Eavesdroppers can be excluded by means of classical cryptography!
Conceptual Approach
Two-Phase Approach

- **Invitation phase**: Discovery of peers, agreement on session parameters
  - Provide meaningful context to subsequent computation

- **Computation phase**: Actual scheduling by means of secure MPC protocol
**Computation Phase**

- iCalendar files as common interface towards the user/external calendar software
- Bit arrays represent free/busy time slots internally

What is actually computed here?
### Scheduling Schemes

- **First match**: Return the first time slot where every participant is available

  ![First Match Example]

- **Best match**: Return the time slot where the most participants are available

  ![Best Match Example]

- **Weighted best match**: Return the time slot with the highest score, where a free time slot contributes with the participant’s assigned priority

  ![Weighted Best Match Example]
Implementation
Programming Language Choice

Development of a fully functional prototype application in Java

• Fast prototyping with clean and robust code
• Extensive standard library for GUI, networking, …
• Available libraries for MPC and iCalendar parsing
High-Level Components

Graphical user interface

Invitation management

iCalendar converter

Circuit generator

MPC black box

Network layer
MPC Black Box

SCAPI (“Secure Computation API”)

- General purpose library that implements cryptographic building blocks for MPC
- Written originally in Java, ported to C++ for performance
- Includes GMW implementation
  - Provides security in semi-honest setting
  - No high-level language that compiles into circuits
  → Boolean circuits need to be constructed by hand
Design easy-to-use circuit API on top of SCAPI

- **Circuit builder**: Encapsulates internal circuit representation of MPC framework
- **Circuit generator**: Expresses the abstract algorithm in boolean logic

```java
class SimpleGenerator implements CircuitGenerator {
    @Override
    public void generate(CircuitBuilder builder) {
        Wire inputA = builder.input(0);
        Wire inputB = builder.input(1);
        Wire result = builder.and(inputA, inputB);
        builder.output(result, 0);
        builder.output(result, 1);
    }
}
```

Recording of logic instead of immediate execution
Scheduling schemes modeled as boolean circuits for secure evaluation

- Formalization of basic building blocks
  - Fixed-length integer operations (addition, comparison, constant multiplication)
  - Bitwise operations (conjunction, disjunction, complement, multiplexing)
  - Filtering

- Depth-efficient aggregation
Invitation Management and Networking

Network layer enables interaction between users

- Sending and receiving invitations
- Connecting to common session

Simple protocol for state synchronization

- Initiator is central mediator
- Each party sends state to initiator in regular intervals
  \[ \rightarrow \text{Synchronize party names, readiness status, …} \]
Derive input bits automatically from personal timetable

→ Find overlaps of time slot candidates with events in calendar file
Live Demo
Evaluation
How long does it take for all parties to successfully schedule an appointment?

- Responsiveness directly perceived by the user
- Circuit depth is determining factor

Approx. $O(n \log n)$ for fixed $m$
How does network latency affect running time?

- Measurements with simulated round-trip time of 100 ms

Running time scaled with constant factor
Network Utilization

How much data is transferred over the network?

- Number of AND gates in the circuit is determining factor

Approx. $O(m \cdot n^3)$
**Usability**

- Participants list

---

Interactive Private Multi-Party Calendar Scheduling
Jonas Nagy-Kuhlen
Informatik 5 Information Systems, Lehrstuhl Prof. Decker
Usability

- Tool for marking time slots as free or unavailable
• Textual appointment description

A public discussion of new cryptographic techniques that allow to maintain privacy in data science.

All times displayed in Europe/Berlin
## Fulfillment of Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Fulfilled</th>
<th>General Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privacy</td>
<td>✓</td>
<td>GMW protocol for secure computation in a semi-honest setting</td>
</tr>
<tr>
<td>Scalability</td>
<td>✓</td>
<td>Use of SCAPI as a low-level framework and optimization of boolean circuits</td>
</tr>
<tr>
<td>Usability</td>
<td>✓</td>
<td>Intuitive graphical user interface</td>
</tr>
<tr>
<td>Workflow integration</td>
<td>✓</td>
<td>iCalendar interoperability</td>
</tr>
<tr>
<td>Flexibility</td>
<td>✓</td>
<td>Easy-to-use API providing general-purpose building blocks for modeling functionality as boolean circuits</td>
</tr>
</tbody>
</table>
Conclusion
Conclusion

- Shown that appointment scheduling can be performed in a privacy-preserving manner without sacrificing functionality and usability
- Developed a fully functional prototype as proof that privacy enhancing building blocks can solve real-world problems
- Demonstrated scalability in practical settings with large numbers of participants
- Implemented powerful API for straightforward construction of boolean circuits
Reusable Software Contributions

• Toolset for boolean circuit generation from general-purpose building blocks
• Generic network protocol for peer discovery and state synchronization

→ Presented approach also applicable for other use cases like voting, auctions, …
Thank you for your attention.
Why use SCAPI as MPC library?

• GMW protocol is secure in semi-honest setting
  – Security against malicious adversaries comes with additional costs
• Boolean circuits considered more flexible than arithmetic circuits in this use case
  – Only small fixed-size integer logic required
• Well-documented and well-structured
  – Can be easily integrated into Java code

“Secure Multi-Party Computation of Boolean Circuits with Applications to Privacy in On-Line Marketplaces”, Choi et al., 2012
First match

- Intersection of free time slots represented by bitwise conjunction
- Subsequent filtering to disclose only first matching index
Weighted best match

- Integer addition allows to accumulate scores for each time slot
- Comparison and multiplexing blocks allow to find index with highest score

Number of parties $n = 4$
Number of time slots $m = 3$
Multiplicative depth of circuit is determining factor for overall running time

- XOR gates can be evaluated locally “for free”
- Independent AND gates are processed in parallel

Approx. linear for fixed $m$