

# Short, Undeniable Signatures for Android 2.x

## Master Semester Project

Sebastien Duc

EPFL

June 15, 2012

# Outline

## 1 Introduction

# Outline

## 1 Introduction

## 2 Signature Schemes

- Undeniable Signatures
- MOVA

# Outline

## 1 Introduction

## 2 Signature Schemes

- Undeniable Signatures
- MOVA

## 3 The Application

- Design
- Application Result

# Outline

- 1 Introduction
- 2 Signature Schemes
  - Undeniable Signatures
  - MOVA
- 3 The Application
  - Design
  - Application Result
- 4 Conclusion

# Outline

## 1 Introduction

## 2 Signature Schemes

- Undeniable Signatures
- MOVA

## 3 The Application

- Design
- Application Result

## 4 Conclusion

# Introduction (1)

# Introduction (1)

- MOVA is an undeniable signature scheme.



# Introduction (1)

- MOVA is an undeniable signature scheme.
- It can achieve very small signatures.

# Introduction (1)

- MOVA is an undeniable signature scheme.
- It can achieve very small signatures.
- Shortness of signatures is very convenient in mobile applications.

# Introduction (1)

- MOVA is an undeniable signature scheme.
- It can achieve very small signatures.
- Shortness of signatures is very convenient in mobile applications.
- The aim of the project was to design an application for Android using MOVA.

# Introduction (2)

## Introduction (2)

- Today there exists a lot of application for classical signatures.

## Introduction (2)

- Today there exists a lot of application for classical signatures.
- It is not the case for undeniable signatures (and MOVA).

## Introduction (2)

- Today there exists a lot of application for classical signatures.
- It is not the case for undeniable signatures (and MOVA).
- We tried to find an application where undeniable signatures can bring something.

# Outline

## 1 Introduction

## 2 Signature Schemes

- Undeniable Signatures
- MOVA

## 3 The Application

- Design
- Application Result

## 4 Conclusion



# Outline

- 1 Introduction
- 2 Signature Schemes
  - Undeniable Signatures
  - MOVA
- 3 The Application
  - Design
  - Application Result
- 4 Conclusion

# Specifications

# Specifications

- Undeniable Signatures are not universally verifiable.

# Specifications

- Undeniable Signatures are not universally verifiable.
- The verifier must run an interactive protocol with the signer

# Definition

Consider two participants  $S$  and  $V$ .  
We define



# Definition

Consider two participants  $S$  and  $V$ .

We define

**Setup**  $(k_p^S, k_s^S) \leftarrow \text{Setup}^S(1^n)$  and  
 $(k_p^V, k_s^V) \leftarrow \text{Setup}^V(1^n)$ .

# Definition

Consider two participants  $S$  and  $V$ .

We define

**Setup**  $(k_p^S, k_s^S) \leftarrow \text{Setup}^S(1^n)$  and  
 $(k_p^V, k_s^V) \leftarrow \text{Setup}^V(1^n)$ .

**Sign**  $\sigma \leftarrow \text{Sign}(m, k_s^S)$

# Definition

Consider two participants  $S$  and  $V$ .

We define

**Setup**  $(k_p^S, k_s^S) \leftarrow \text{Setup}^S(1^n)$  and  
 $(k_p^V, k_s^V) \leftarrow \text{Setup}^V(1^n)$ .

**Sign**  $\sigma \leftarrow \text{Sign}(m, k_s^S)$

**Confirm** Interactive protocol between  $S$  and  $V$  to confirm the validity of  $(m, \sigma)$ .



# Definition

Consider two participants  $S$  and  $V$ .

We define

**Setup**  $(k_p^S, k_s^S) \leftarrow \text{Setup}^S(1^n)$  and  
 $(k_p^V, k_s^V) \leftarrow \text{Setup}^V(1^n)$ .

**Sign**  $\sigma \leftarrow \text{Sign}(m, k_s^S)$

**Confirm** Interactive protocol between  $S$  and  $V$  to confirm the validity of  $(m, \sigma)$ .

**Deny** Interactive protocol between  $S$  and  $V$  to deny the validity of  $(m, \sigma')$ .

# Outline

- 1 Introduction
- 2 Signature Schemes
  - Undeniable Signatures
  - MOVA
- 3 The Application
  - Design
  - Application Result
- 4 Conclusion

# MOVA Signature Scheme

# MOVA Signature Scheme

- MOVA is a scheme for undeniable, short signatures.

# MOVA Signature Scheme

- MOVA is a scheme for undeniable, short signatures.
- Provides batch verification.

# MOVA Signature Scheme

- MOVA is a scheme for undeniable, short signatures.
- Provides batch verification.
- Scheme based on group homomorphism.

# Setup

Consider a pseudo-random generator  $Gen_K$ .

# Setup

Consider a pseudo-random generator  $GenK$ .

- 1 choose two groups  $G, H$ .
- 2 choose a homomorphism  $h : G \rightarrow H$
- 3 Generate  $Xkeys \leftarrow GenK(seedK)$ ,  $Xkeys \in G^{L_{key}}$ .
- 4 Compute  $Ykeys = h^{L_{key}}(Xkeys)$ .



# Setup

Consider a pseudo-random generator  $GenK$ .

- 1 choose two groups  $G, H$ .
- 2 choose a homomorphism  $h : G \rightarrow H$
- 3 Generate  $Xkeys \leftarrow GenK(seedK)$ ,  $Xkeys \in G^{L_{key}}$ .
- 4 Compute  $Ykeys = h^{L_{key}}(Xkeys)$ .

Public key:  $(G, H, |H|, seedK, Ykeys)$

Secret key:  $h$

# Signature

Consider a message  $m \in \{0,1\}^*$  and the pseudo-random generator  $GenS$ .

# Signature

Consider a message  $m \in \{0,1\}^*$  and the pseudo-random generator  $GenS$ .

- 1 Generate  $Xsigs \leftarrow GenS(m)$ , where  $Xsigs \in G^{Lsig}$
- 2 Compute  $Ysigs = h^{Lsig}(Xsigs)$ .

# Signature

Consider a message  $m \in \{0, 1\}^*$  and the pseudo-random generator  $GenS$ .

- 1 Generate  $Xsigs \leftarrow GenS(m)$ , where  $Xsigs \in G^{Lsig}$
- 2 Compute  $Ysigs = h^{Lsig}(Xsigs)$ .

Signature:  $Ysigs$

# Group Interpolation

## Definition

We say the  $S \subseteq G \times H$  interpolates in a group homomorphism if  $\exists$  a homomorphism  $h$  st.  $h(x) = y \ \forall (x, y) \in S$ .

Note: In MOVA we consider sets  $S$  that interpolates in a unique group homomorphism.

# Verification

The verification is an interactive protocol between  $S$  and  $V$ .

# Verification

The verification is an interactive protocol between  $S$  and  $V$ .

The prover convinces the verifier that

$$\{(Xkey_i, Ykey_i) | i = 1, \dots, Lkey\} \cup \{(Xsig_i, Ysig_i) | i = 1, \dots, Lsig\}$$

interpolates in a group homomorphism.

# Batch Verification



# Batch Verification

- Used to verify multiple signature.

# Batch Verification

- Used to verify multiple signature.
- Signatures must be issued from same key pair.

# Batch Verification

- Used to verify multiple signature.
- Signatures must be issued from same key pair.
- Verify all the signatures in only one protocol call.

# Outline

- 1 Introduction
- 2 Signature Schemes
  - Undeniable Signatures
  - MOVA
- 3 The Application**
  - Design
  - Application Result
- 4 Conclusion

# Motivation

# Motivation

- The application is a University Contest

# Motivation

- The application is a University Contest
- The concept is similar to the University Challenge in the UK, but using phones.

# Outline

- 1 Introduction
- 2 Signature Schemes
  - Undeniable Signatures
  - MOVA
- 3 The Application
  - Design
  - Application Result
- 4 Conclusion



# Overview (1)

# Overview (1)

- Universities are providing quizzes to teams in other universities.

# Overview (1)

- Universities are providing quizzes to teams in other universities.
- Teams win points by answering quizzes correctly and this increased the final score of their university.

# Overview (1)

- Universities are providing quizzes to teams in other universities.
- Teams win points by answering quizzes correctly and this increased the final score of their university.
- The university with the highest score wins the contest.

# Overview (2)

# Overview (2)

- Quizzes are uniquely assigned to the teams.

# Overview (2)

- Quizzes are uniquely assigned to the teams.
- When subscribing, teams choose a password for future authentication.

## Overview (2)

- Quizzes are uniquely assigned to the teams.
- When subscribing, teams choose a password for future authentication.
- Quizzes can contain normal or multiple choice questions.



# Overview (2)

- Quizzes are uniquely assigned to the teams.
- When subscribing, teams choose a password for future authentication.
- Quizzes can contain normal or multiple choice questions.
- Correction and scoring is done manually by the server manager.

# Setup

Consider two universities  $U_1$ ,  $U_2$  and a team  $T_i$  in university  $U_2$ .

# Setup

Consider two universities  $U_1$ ,  $U_2$  and a team  $T_i$  in university  $U_2$ .

- 1 Server  $S_1$  of  $U_1$  provides quizzes to  $T_i$  with a MOVA signature.

# Setup

Consider two universities  $U_1$ ,  $U_2$  and a team  $T_i$  in university  $U_2$ .

- 1 Server  $S_1$  of  $U_1$  provides quizzes to  $T_i$  with a MOVA signature.
- 2 When filled-in, the quiz is sent to server  $S_2$  of  $U_2$ .

# Setup

Consider two universities  $U_1$ ,  $U_2$  and a team  $T_i$  in university  $U_2$ .

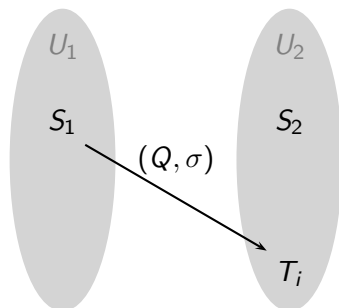
- 1 Server  $S_1$  of  $U_1$  provides quizzes to  $T_i$  with a MOVA signature.
- 2 When filled-in, the quiz is sent to server  $S_2$  of  $U_2$ .
- 3  $S_2$  signs it and send it back to  $T_i$  with the signature.

# Setup

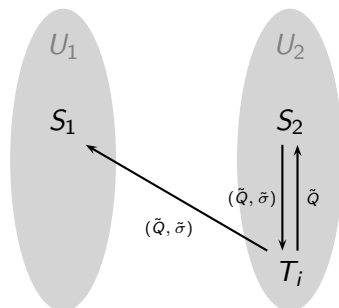
Consider two universities  $U_1$ ,  $U_2$  and a team  $T_i$  in university  $U_2$ .

- 1 Server  $S_1$  of  $U_1$  provides quizzes to  $T_i$  with a MOVA signature.
- 2 When filled-in, the quiz is sent to server  $S_2$  of  $U_2$ .
- 3  $S_2$  signs it and send it back to  $T_i$  with the signature.
- 4  $T_i$  can verify the signature and send it to  $S_1$  with the signature.

## Sending Quiz



## Sending Quiz Back



# Security



# Security

- Every signature is done with MOVA.

# Security

- Every signature is done with MOVA.
- Quizzes are signed so that teams are ensured not to have a fake one.

# Security

- Every signature is done with MOVA.
- Quizzes are signed so that teams are ensured not to have a fake one.
- Servers use batch verification.

# Security

- Every signature is done with MOVA.
- Quizzes are signed so that teams are ensured not to have a fake one.
- Servers use batch verification.
- Teams are authenticated to their respective university server when sending the quizzes back.

# Security

- Every signature is done with MOVA.
- Quizzes are signed so that teams are ensured not to have a fake one.
- Servers use batch verification.
- Teams are authenticated to their respective university server when sending the quizzes back.
- Authentication is done using a simple challenge-response protocol.

# Threat Model

An adversary could try the following

# Threat Model

An adversary could try the following

- Forge fake quizzes. Hence the signature when sending quizzes.

# Threat Model

An adversary could try the following

- Forge fake quizzes. Hence the signature when sending quizzes.
- Fill in the quizzes of another team. Not possible by construction.



# Threat Model

An adversary could try the following

- Forge fake quizzes. Hence the signature when sending quizzes.
- Fill in the quizzes of another team. Not possible by construction.
- Modify the answers in a filled-in quiz. Hence the signature when sending back.

# Implementation of MOVA

# Implementation of MOVA

- $G = (\mathbb{Z}/n\mathbb{Z})^*$  where  $n = pq$  is the product of two large primes.

# Implementation of MOVA

- $G = (\mathbb{Z}/n\mathbb{Z})^*$  where  $n = pq$  is the product of two large primes.
- $H = \{-1, 1\}$ .

# Implementation of MOVA

- $G = (\mathbb{Z}/n\mathbb{Z})^*$  where  $n = pq$  is the product of two large primes.
- $H = \{-1, 1\}$ .
- The homomorphism can be either  $\left(\frac{\cdot}{p}\right)$  or  $\left(\frac{\cdot}{q}\right)$ .

# Outline

- 1 Introduction
- 2 Signature Schemes
  - Undeniable Signatures
  - MOVA
- 3 The Application
  - Design
  - Application Result
- 4 Conclusion

# Client

Android Application. It has three main activities.

# Client

Android Application. It has three main activities.

**ChallengeActivity** To get the latest quiz.  
Answer to questions and send it back.



# Client

Android Application. It has three main activities.

**ChallengeActivity** To get the latest quiz.  
Answer to questions and send it back.

**UniversityScoreActivity** To get the score of all participating universities.

# Client

Android Application. It has three main activities.

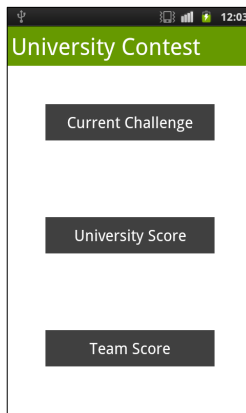
**ChallengeActivity** To get the latest quiz.  
Answer to questions and send it back.

**UniversityScoreActivity** To get the score of  
all participating universities.

**TeamScoreActivity** To get his team score  
and the results of the quizzes.



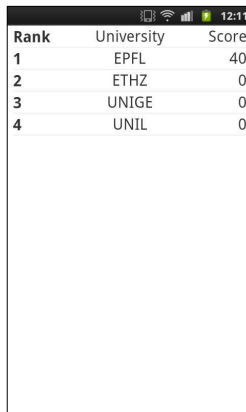
# Main Menu



# ChallengeActivity

The screenshot shows a mobile application interface for a 'University Contest'. At the top, there's a status bar with a USB icon, signal strength, battery, and the time 12:04. Below the status bar is a green header with the title 'University Contest'. Underneath the header is a green bar labeled 'Current Challenge' with a 'Due To: 2012-06-29' timestamp. The main content area has a dark grey header for 'Question' followed by the text 'What is the full name of gauss?'. Below this is another dark grey header for 'Answer' followed by a text input field with a blue border. The next section is another 'Question' header with the text 'When was euler born?'. Below this is an 'Answer' header followed by three radio button options: 'in 1606', 'in 1707', and 'in 1808'. At the bottom of the screen is a 'Send' button.

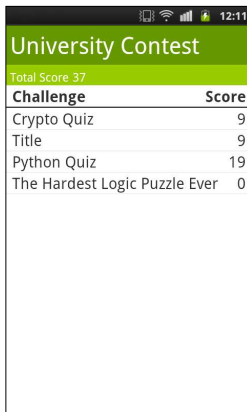
# UniversityScoreActivity



The screenshot shows an Android application interface. At the top, there is a status bar with icons for signal, Wi-Fi, battery, and the time 12:11. Below the status bar is a table with three columns: Rank, University, and Score. The table contains four rows of data. The first row shows Rank 1, University EPFL, and Score 40. The second row shows Rank 2, University ETHZ, and Score 0. The third row shows Rank 3, University UNIGE, and Score 0. The fourth row shows Rank 4, University UNIL, and Score 0. The table is displayed on a white background with black text. The status bar is black with white icons.

Rank	University	Score
1	EPFL	40
2	ETHZ	0
3	UNIGE	0
4	UNIL	0

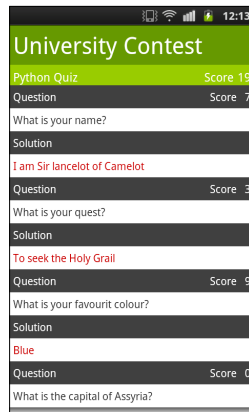
# TeamScoreActivity



University Contest

Total Score 37

Challenge	Score
Crypto Quiz	9
Title	9
Python Quiz	19
The Hardest Logic Puzzle Ever	0



University Contest

Python Quiz Score 19

Question	Score
What is your name?	7
Solution	
I am Sir lancelot of Camelot	
Question	Score 3
What is your quest?	
Solution	
To seek the Holy Grail	
Question	Score 9
What is your favourit colour?	
Solution	
Blue	
Question	Score 0
What is the capital of Assyria?	

# Server

Two Java applications and a MySQL database.

# Server

Two Java applications and a MySQL database.

**Query Handler Process** A constantly alive thread listening to a specified port and answering to queries.



# Server

Two Java applications and a MySQL database.

**Query Handler Process** A constantly alive thread listening to a specified port and answering to queries.

**Server Manager** An application used to manage the server.  
Manage quizzes, teams and universities.

# Server

Two Java applications and a MySQL database.

**Query Handler Process** A constantly alive thread listening to a specified port and answering to queries.

**Server Manager** An application used to manage the server.  
Manage quizzes, teams and universities.

**unicontest** The database consisting of three tables  
(team, challenge and university).

# Challenge Manager

University Server Contest Manager

Team Management   University Management   **Challenge Management**

Choose action **new challenge**

## New challenge

Challenge title

Assigned team

Due to

Question 1

Solution 1

Question 2

☐ A basketball

☐ A football

☒ A shrubbery

☐ A chicken

☐ A pig

MCQ

# Outline

- 1 Introduction
- 2 Signature Schemes
  - Undeniable Signatures
  - MOVA
- 3 The Application
  - Design
  - Application Result
- 4 Conclusion

# Conclusion

- We have designed an application for android using MOVA.
- MOVA serves such real-life mobile apps.: e.g., by its shortness and by the batch verification.

# Conclusion

- We have designed an application for android using MOVA.
- MOVA serves such real-life mobile apps.: e.g., by its shortness and by the batch verification.

Future Work: Find other applications using MOVA which look a bit less artificial.

# Thanks for your attention !