Monitoring Electronic Exams

Ali Kassem\textsuperscript{1}, Yliès Falcone\textsuperscript{2} and Pascal Lafourcade\textsuperscript{3}

\textsuperscript{1}Univ. Grenoble Alpes, VERIMAG, Grenoble, France
\textsuperscript{2}Univ. Grenoble Alpes, Inria, LIG, Grenoble
\textsuperscript{3}Université Clermont Auvergne, LIMOS, France

The 15th International Conference on Runtime Verification

Vienna, September 28, 2015
Traditional Exam
Information technology for the assessment of knowledge and skills.
Threats...

- Candidate cheating
- Bribed, corrupted or unfair examiners
- Dishonest/untrusted exam authority
- Outside attackers
- ...
Most existing e-exam systems assume trusted authorities and focus on student cheating:

- Exam centers
- Software solutions, e.g. ProctorU
... and their Mitigation

Most existing e-exam systems assume trusted authorities and focus on student cheating:

- Exam centers
- Software solutions, e.g. ProctorU

Yet also the other threats are real:

- Atlanta Public Schools cheating scandal (2009)
- UK student visa tests fraud (2014)
Most existing e-exam systems assume **trusted authorities** and focus on **student cheating**:

- Exam centers
- Software solutions, e.g. ProctorU

Yet also the **other threats** are real:

- Atlanta Public Schools cheating scandal (2009)
- UK student visa tests fraud (2014)

So what about **dishonest authorities** or **hackers**?
Several Security Properties

Secrypt’14  **Authentication Properties:** Mark Authenticity, Answer Origin Authentication, Form Authorship, Form Authenticity.

**Privacy Properties:** Anonymous Marking, Question Indistinguishability, Anonymous Examiner, Mark Privacy, Mark Anonymity

ISPEC’15  **Individual Verifiability:** Question Validity, Marking Correctness, Exam-Test Integrity, Exam-Test Markedness, Marking Integrity, Marking Notification Integrity

**Universal Verifiability:** Eligibility (Registration), Marking Correctness Exam-Test Integrity, Exam-Test Markedness, Marking Integrity.
Several Security Properties

Secrypt’14 Authentication Properties: Mark Authenticity, Answer Origin Authentication, Form Authorship, Form Authenticity.
Privacy Properties: Anonymous Marking, Question Indistinguishability, Anonymous Examiner, Mark Privacy, Mark Anonymity

ISPEC’15 Individual Verifiability: Question Validity, Marking Correctness, Exam-Test Integrity, Exam-Test Markedness, Marking Integrity, Marking Notification Integrity
Universal Verifiability: Eligibility (Registration), Marking Correctness Exam-Test Integrity, Exam-Test Markedness, Marking Integrity.

How can we use it on real e-exam?
Plan

Introduction

Model

Properties

Case Study: UJF E-exam

Conclusion
Plan

Introduction

Model

Properties

Case Study: UJF E-exam

Conclusion
E-exam: Players and Organization

Three Roles:

Candidate

Examination Authority

Examiner
E-exam: Players and Organization

Three Roles:

Candidate

Examination Authority

Examiner

Four Phases:

Event Based Model

1. Registration
   Register

2. Examination
   begin
   get
   Question
   change
   submit
   accept
   Answer
   end
Event Based Model

1. Registration
Event Based Model

1. Registration

Register

register( )
Event Based Model

1. Registration
   Register
   $register(\text{student})$

2. Examination
Event Based Model

1. Registration
   - Register

2. Examination
   - $begin()$

- $register()$
Event Based Model

1. Registration
   Register
   \[ \text{register}(\text{\textbullet}) \]

2. Examination
   begin
   \[ \text{begin}(\text{\textbullet}) \]
   get
   \[ \text{get}(\text{\textbullet}, \text{?}) \]
   Question
Event Based Model

1. Registration
   - Register
   - \( \text{register}(\text{•}) \)

2. Examination
   - \( \text{begin}() \)
   - \( \text{get}() \)
   - \( \text{change}() \)
Event Based Model

1. Registration

2. Examination

begin

get

change

submit

register

Question

Answer

accept
Event Based Model

1. Registration
   - Register
   - register

2. Examination
   - begin
   - get
   - change
   - submit
   - end
   - Question
   - Answer
   - accept
Event Based Model

3. Marking

Correct Answer

Evaluation

Mark 12 / 30
Event Based Model

3. Marking

\[ corr(?, \checkmark) \]

Correct Answer
3. Marking

$corr(?,\checkmark)$

Correct Answer

Evaluation

$mark(?,\ ?,\ !,\ T/F)$
Event Based Model

3. Marking

\[ corr(\text{?}, \text{✓}) \]
Correct Answer

Evaluation

\[ mark(\text{!}, \text{?}, \text{!}, \text{TF}) \]

4. Notification
Event Based Model

3. Marking

4. Notification

Correct Answer

Evaluation

Mark

Assign

corr(?, ✓)

mark(, ?, !, TF)

assign(, A+)
Plan

Introduction

Model

Properties

Case Study: UJF E-exam

Conclusion
Quantified Event Automata (QEAs)

- Properties expressed as **QEAs** [BFH+12]: event automaton with quantified variables.

- An event automaton is a **finite-state machine** with transitions labeled by parametric events.

- Transitions may include **guards** and **assignments**.

- We extend the initial definition of QEAs by:
  1. variable declaration and **initialization** before reading the trace
  2. **global variable** shared among all event automaton instances.

\[
event(\text{parameters}) \frac{[\text{guard}]}{\text{assignment}}
\]
Candidate Eligibility

No answer is accepted from an unregistered candidate

\[ \Sigma = \{ \text{register}(i), \text{accept}(i, q, a) \} \]

\[ \forall i \]

![Diagram showing a transition from 1 to 2 labeled with register(i)]
Candidate Eligibility

No answer is accepted from an unregistered candidate

\[ \forall i \text{ register}(i) \]

\[ \Sigma = \{ \text{register}(i), \text{accept}(i, q, a) \} \]
Candidate Eligibility with Auditing

All candidates that violates the requirement are collected in a set $F$.

Initially: $I : \triangleq \emptyset$

- **Register** $(i) \quad I := I \cup \{i\}$
- **Accept** $(i, q, a) \quad [i \notin I] \quad F := F \cup \{i\}$
Candidate Registration: an unregistered candidate tried to take the exam.
Properties

Candidate Registration: an unregistered candidate tried to take the exam.

Answer Authentication:
- an unsubmitted answer was considered as accepted; or
- more than one answer were accepted from a candidate.
Candidate Registration: an unregistered candidate tried to take the exam.

Answer Authentication:
- an unsubmitted answer was considered as accepted; or
- more than one answer were accepted from a candidate.

Questions Ordering:
- a candidate got a question before validating the previous ones.
Exam Availability: an answer was accepted outside exam time.
Exam Availability: an answer was accepted outside exam time.

Exam Availability with Flexibility:

- supports different duration and starting time between candidates.
Exam Availability: an answer was accepted outside exam time.

Exam Availability with Flexibility:

- supports different duration and starting time between candidates.

Marking Correctness: an answer was marked in a wrong way.
Exam Availability: an answer was accepted outside exam time.

Exam Availability with Flexibility:
- supports different duration and starting time between candidates.

Marking Correctness: an answer was marked in a wrong way.

Mark Integrity:
- an accepted answer was not marked; or
- a candidate was not assigned the corresponding mark.
Plan

Introduction

Model

Properties

Case Study: UJF E-exam

Conclusion
Registration:

- 2 weeks before the exam.
- Using login/password.
Examination in a supervised room

Authentication and answers questions as follows:

▶ In a fixed order.

▶ Once validates the current question, he gets the next one.

▶ He can change the answer unlimited times before validating.

▶ Once he validates, then he cannot go back and change any of the validated answers.
Marking:

- For each question, the professor specifies the correct answer(s).
- For each question, all the answers provided by the candidates are collected.
- Each answer is evaluated by an examiner to 0 or 1.
- The mark for each candidate is calculated as the summation of all the scores attributed to his answers.

Notification:

- The marks are notified to the candidates.
- A candidate can consult his submission and check the marking.
Analysis

Verification of two real e-exam executions using MarQ tool [RCR15].

From the logs: $\text{register}(i)$, $\text{change}(i, q, a)$, $\text{submit}(i, q, a)$, $\text{accept}(i, q, a)$.

4 Properties

- Candidate Registration
- Candidate Eligibility
- Answer Authentication
- Exam Availability
5 new properties

- **Answer Authentication**: 
  - All accepted answers are submitted by candidates.
  - Allow the acceptance of the same answer again.
  - But, still forbids the acceptance of a different answer.
5 new properties

- **Answer Authentication**: All accepted answers are submitted by candidates. Allow the acceptance of the same answer again. But, still forbids the acceptance of a different answer.

- **Answer Authentication Reporting**: Collects in a set $F$ every candidate from which more than one answer are accepted.
5 new properties

- **Answer Authentication**: All accepted answers are submitted by candidates. Allow the acceptance of the same answer again. But, still forbids the acceptance of a different answer.

- **Answer Authentication Reporting**: Collects in a set $F$ every candidate from which more than one answer are accepted.

- **Answer Editing**: A candidate cannot change an answer after validation it.
5 new properties

- **Answer Authentication**: All accepted answers are submitted by candidates. Allow the acceptance of the same answer again. But, still forbids the acceptance of a different answer.

- **Answer Authentication Reporting**: Collects in a set $F$ every candidate from which more than one answer are accepted.

- **Answer Editing**: A candidate cannot change an answer after validation it.

- **Question Ordering**: A candidate cannot changes the answer to a future question before validating the current question.
5 new properties

- **Answer Authentication**: All accepted answers are submitted by candidates. 
  - **Allow the acceptance of the same answer again**.
  - **But, still forbids the acceptance of a different answer**.
- **Answer Authentication Reporting**: Collects in a set $F$ every candidate from which more than one answer are accepted.
- **Answer Editing**: A candidate cannot change an answer after validation it.
- **Question Ordering**: A candidate cannot change the answer to a future question before validating the current question.
- **Acceptance Order**: A candidate has to validate the questions in order, but he can skip some questions.
## Results: Exam 1

233 students, 40875 events

<table>
<thead>
<tr>
<th>Property</th>
<th>Result</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate Registration</td>
<td>✓</td>
<td>538</td>
</tr>
<tr>
<td>Candidate Eligibility</td>
<td>✓</td>
<td>517</td>
</tr>
<tr>
<td>Answer Authentication</td>
<td>×</td>
<td>310</td>
</tr>
<tr>
<td>Exam Availability</td>
<td>✓</td>
<td>518</td>
</tr>
<tr>
<td>Answer Authentication *</td>
<td>✓</td>
<td>742</td>
</tr>
<tr>
<td>Answer Authentication Reporting</td>
<td>× [1]</td>
<td>654</td>
</tr>
<tr>
<td>Answer Editing</td>
<td>✓</td>
<td>641</td>
</tr>
<tr>
<td>Question Ordering *</td>
<td>×</td>
<td>757</td>
</tr>
<tr>
<td>Acceptance Order</td>
<td>✓</td>
<td>697</td>
</tr>
</tbody>
</table>
Results: Exam 2

90 students, 4641 events

<table>
<thead>
<tr>
<th>Property</th>
<th>Result</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate Registration</td>
<td>✓</td>
<td>230</td>
</tr>
<tr>
<td>Candidate Eligibility</td>
<td>✓</td>
<td>214</td>
</tr>
<tr>
<td>Answer Authentication</td>
<td>✓</td>
<td>275</td>
</tr>
<tr>
<td>Exam Availability</td>
<td>✗ [1]</td>
<td>237</td>
</tr>
<tr>
<td>Answer Authentication *</td>
<td>✓</td>
<td>223</td>
</tr>
<tr>
<td>Answer Authentication Reporting</td>
<td>✓</td>
<td>265</td>
</tr>
<tr>
<td>Answer Editing</td>
<td>✗</td>
<td>218</td>
</tr>
<tr>
<td>Question Ordering *</td>
<td>✗</td>
<td>389</td>
</tr>
<tr>
<td>Acceptance Order</td>
<td>✓</td>
<td>294</td>
</tr>
</tbody>
</table>
Plan

Introduction

Model

Properties

Case Study: UJF E-exam

Conclusion
Conclusion

- Event-based model of e-exams.
- Several properties defined as QEAs.
- Analysis of 2 real e-exams at UJF using MarQ tool.
- Discovering some misbehaviours.
Future Work

- Analyze more existing e-exams from other universities.
- Perform on-line verification with our monitors during live e-exams.
- Study more expressive and quantitative properties that can detect colluded students through similar answer patterns.
- Automatic transformation from verifiability to monitors.
Thank you for your attention!

Questions?

pascal.lafourcade@udamail.fr
Howard Barringer, Yliès Falcone, Klaus Havelund, Giles Reger, and David E. Rydeheard.
Quantified event automata: Towards expressive and efficient runtime monitors.

Giles Reger, Helena Cuenca Cruz, and David E. Rydeheard.
MarQ: Monitoring at runtime with QEA.
No answer is accepted from an unregistered candidate.
Initially: $I : \triangleq \emptyset$

- $\text{accept}(i, q, a)$ \quad $F \triangleq F \cup \{(i, q, a)\}$

- $\text{register}(i)$ \quad $I \triangleq I \cup \{i\}$

- $\text{register}(i)$ \quad $I \triangleq I \cup \{i\}$
Answer Authentication

- All accepted answers are submitted by candidates.
- Exactly one answer is accepted from each candidate.
A candidates can take the exam only during the examination time.

\[
\Sigma_{EA}(i, t) \begin{cases} [t_0 \leq t \leq t_f] \\
F \triangleq \{i\} \end{cases}
\]

\[
\Sigma_{EA}(i, t) \begin{cases} [t_0 > t \lor t > t_f] \\
F \triangleq F \cup \{i\} \end{cases}
\]

- \(\Sigma_{EA} = \{get(i, t), change(i, t), submit(i, t), accept(i, t)\}\).
- \(t_0\) is the starting instant of the exam.
- \(t_f\) is the ending instant of the exam.
Exam Availability with Flexibility

Exam Availability with flexible starting time and duration.

∀ $i$

$\begin{align*}
\text{begin}(i, t) & \quad \frac{[t_1 \leq t \leq t_2]}{t_b \triangleq t} \\
\text{accept}(i, t) & \quad [t_b \leq t \leq t_2 \land t - t_b \leq \text{duration}_i]
\end{align*}$

- $t_1$ is the starting instant of the allowed period.
- $t_2$ is the ending instant of the allowed period.
All answers were marked correctly.

\[ \forall q, A : \hat{\exists} \emptyset \]

\[ \text{corrAns}(q, a) \quad \overline{A : \hat{=} A \cup \{a\}} \]

\[ \text{marked}(q, a, b) \quad [(b=1 \iff a \in A)] \]

\[ \text{marked}(q, a, b) \quad [b=1 \iff a \in A] \]
All accepted answers were marked;
each candidate was assigned the mark attributed to his answers.
A weaker variant of Answer Authentication:

- All accepted answers are submitted by candidates.
- Allow the acceptance of the same answer again.
- But, still forbids the acceptance of a different answer.

Motivation: UJF exam allows the acceptance of the same answer twice.
Collects in a set $F$ every candidate from which more than one answer are accepted.

**Global:** $F : \triangleq \emptyset \ \forall q$

- $\text{accept}(i, q, a) \quad \frac{[i \notin A]}{A : \triangleq A \cup \{i\}}$
- $\text{accept}(i, q, a) \quad \frac{[i \notin A]}{A : \triangleq A \cup \{i\}}$
- $\text{accept}(i, q, a) \quad \frac{[i \in A]}{F : \triangleq \{i\}}$
- $\text{accept}(i, q, a) \quad \frac{[i \in A]}{F : \triangleq F \cup \{i\}}$

**Diagram:**

1. $\text{accept}(i, q, a) \quad \frac{[i \notin A]}{A : \triangleq \{i\}}$
2. $\text{accept}(i, q, a) \quad \frac{[i \in A]}{F : \triangleq \{i\}}$
3. $\text{accept}(i, q, a) \quad \frac{[i \in A]}{F : \triangleq F \cup \{i\}}$
A candidate cannot change an answer after validation it.

\[ \forall i, \forall q \]

\[ \text{change}(i, q) \quad \text{accept}(i, q, a) \quad [a=a_v] \]

Motivation: UJF exam does not allow a candidate to change any of the previously validated answers.
A candidate cannot change the answer to a future question before validating the current question.

\[
\begin{align*}
\forall i \quad & \quad \text{change}(i, q) & \quad \text{[ord}(q)=1] \\
& \quad \text{accept}(i, q) & \quad \text{[ord}(q)=1] \quad c: \not=2 \\
\quad & \quad \text{accept}(i, q) & \quad \text{[ord}(q)<c] \\
\quad & \quad \text{accept}(i, q) & \quad \text{[ord}(q)=c] \quad c++ \\
\quad & \quad \text{change}(i, q) & \quad \text{[ord}(q)<c] \\
\quad & \quad \text{change}(i, q) & \quad \text{[ord}(q)=c] \\
\quad & \quad \text{accept}(i, q) & \quad \text{[ord}(q)<c] \\
\quad & \quad \text{accept}(i, q) & \quad \text{[ord}(q)=c] \quad c++
\end{align*}
\]

Motivation: developers did not log anything related to the event \( \text{get}(i, q) \) (needed for Question Ordering).
Acceptance Order

A candidate has to validate the questions in order, but he can skip some questions.

\[
\forall i, c : \hat{=} 1 \Rightarrow \text{accept}(i, q) \quad (\text{ordq} \geq c) \\
\text{c} := \text{ordq}
\]

Motivation: allows us to check if candidates answer the question in lexicographic order when Question Ordering * fails.

It is the case when a candidate able to skip some questions.