Modeling and Synthesizing Privacy-Preserving Applications

Raphaël Mannadiar, Mohamed Layouni, Hans Vangheluwe

McGill University

November 2009

outline

eServices

The Belgian government is currently moving some of its services towards digital mediums.

These *eServices* provide advantages such as higher speed, availability and accessibility.

For these eServices to be widely accepted and adopted by the population, notions of privacy, security and authenticity need to be guaranteed.

An eService Scenario

One example of an eService is "Prescription Issuing".

Premise: A patient P needs a prescription of medicine M.

1. P electronically contacts someone he believes to be doctor D;
2. D authenticates himself as a certified physician without revealing his identity;
3. P authenticates himself as a valid patient (e.g. insured) without revealing his identity;
4. Using P's credentials, D verifies that P should receive M;
5. D writes and securely transmits a prescription to P such that it may only be used n times and only by P;
6. P takes whatever steps are necessary to receive M (e.g. communicate with pharmacist...).

An eService Scenario... drawn

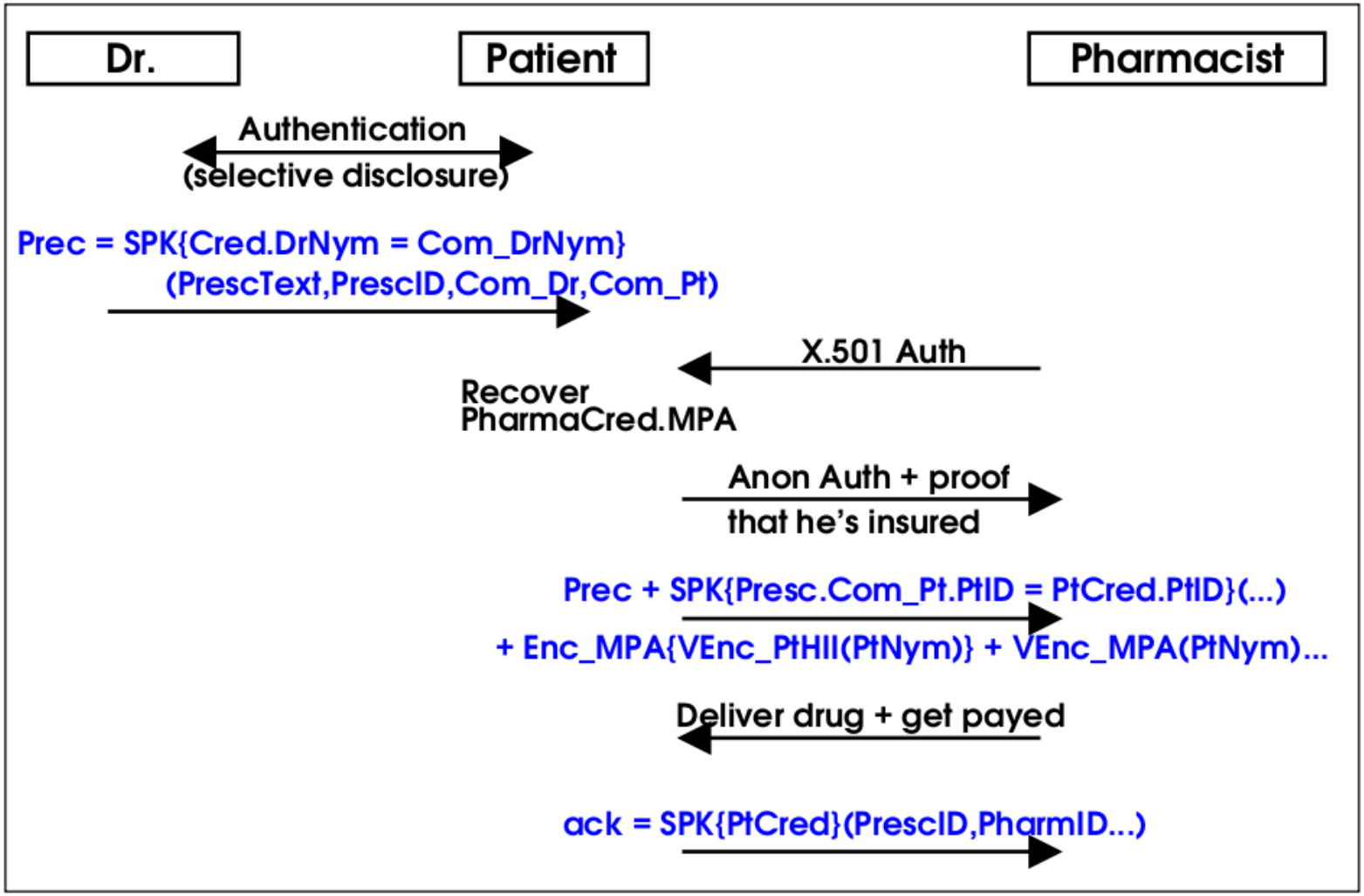


Figure: Prescription Issuing

An eService Scenario... concerns

The problem

What is needed to implement the described prescription issuing system?

A cryptography expert → authentication, encryption, forgery-prevention...

A programming expert → UI, networking...

As is often the case, these two experts might be different people. A few paths can now be taken...

An eService Scenario... implementation techniques

- Solo implementation
  → possibly optimal, probably sub-optimal and flawed solution;
- Coop implementation (Code)
  → 1 satisfactory solution;
- Coop implementation (DSM)
  → environment for modeling and synthesizing any and all privacy-preserving applications.

Modeling vs. Coding

Our approach : DSM or Domain-Specific Modeling.

Its benefits include:

- Use of domain concepts
  → no more mental concept translation;
- Automatic code synthesis from models
  → no more programmer middle-man;
  → faster and more robust development and evolution;
- One model
  → mutliple target platforms;
- Models vs. code
  → easier to understand, simulate and analyze.

An eService Scenario... modeled

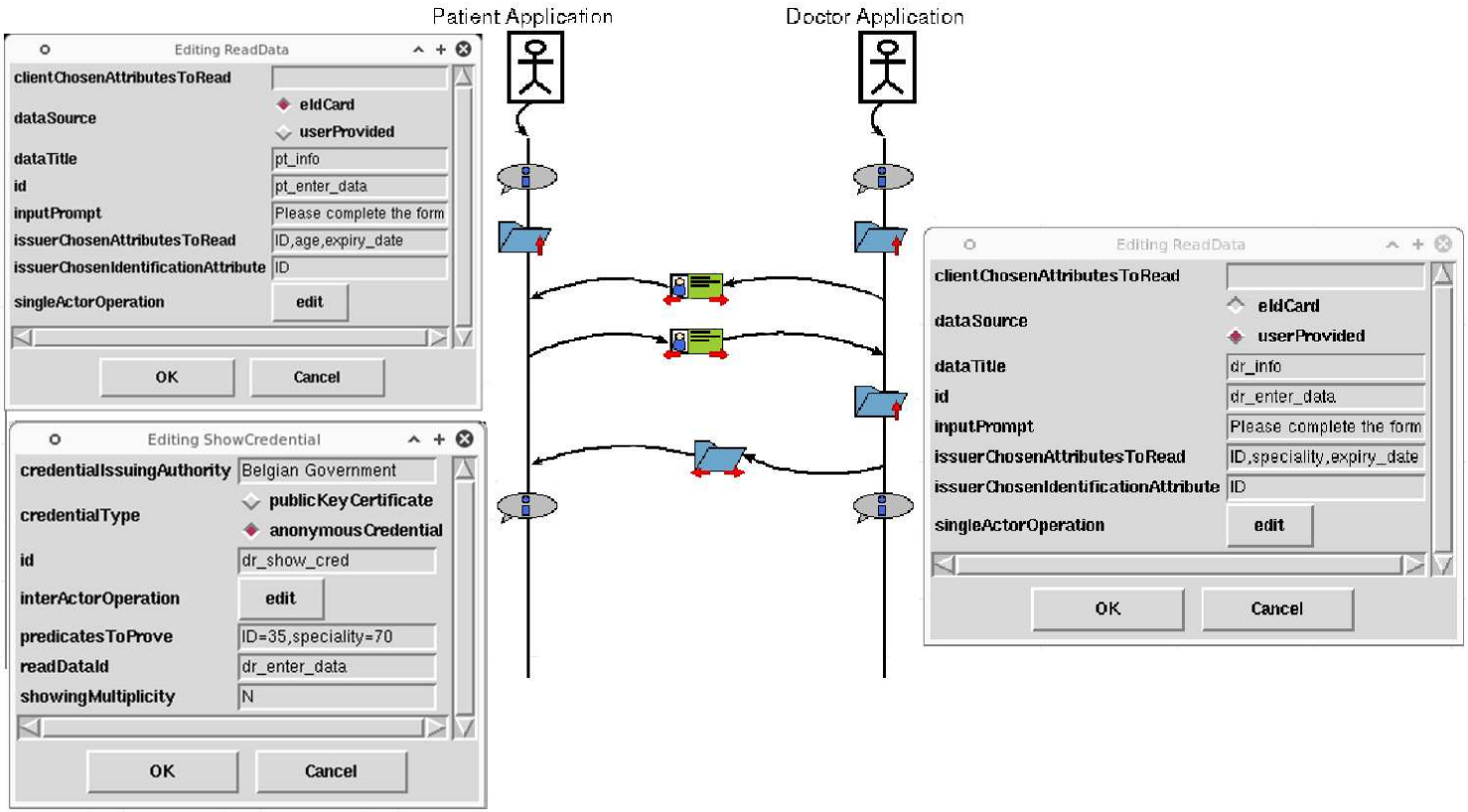


Figure: Prescription Issuing

An eService Scenario... generated

Application synthesis occurs via model transformations from high-level models down to code.

Left-hand side

`InformationMessage` containing text T

$\longrightarrow$

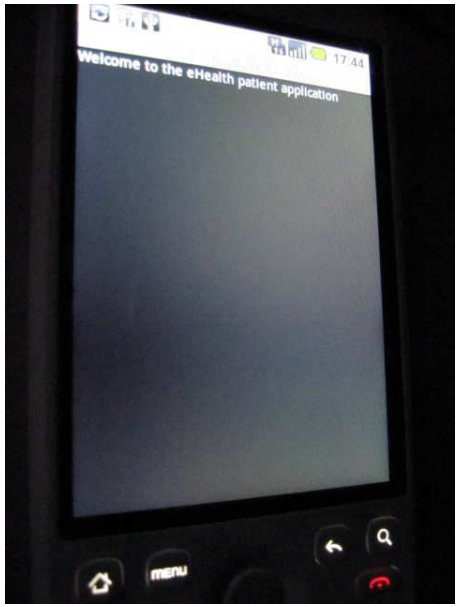
Right-hand side

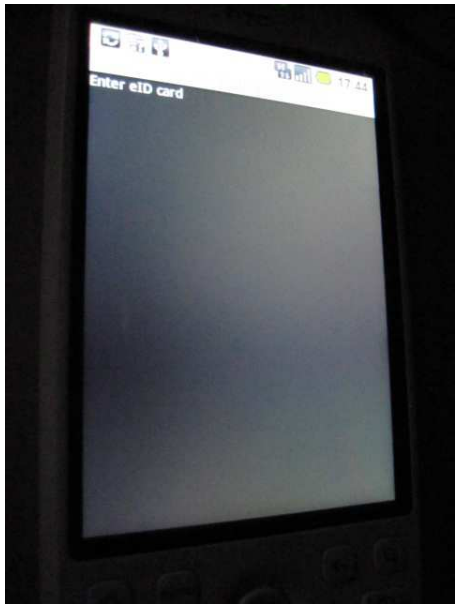
Scrollable, right-aligned screen with text T and "Ok" button that prompts transition to next screen

Figure: An example model transformation rule

The result : 2 applications with
$\longrightarrow$ a Google Android or Internet Browser user interface;
$\longrightarrow$ a remote back-end for complex cryptographic functions.

An eService Scenario... in the flesh





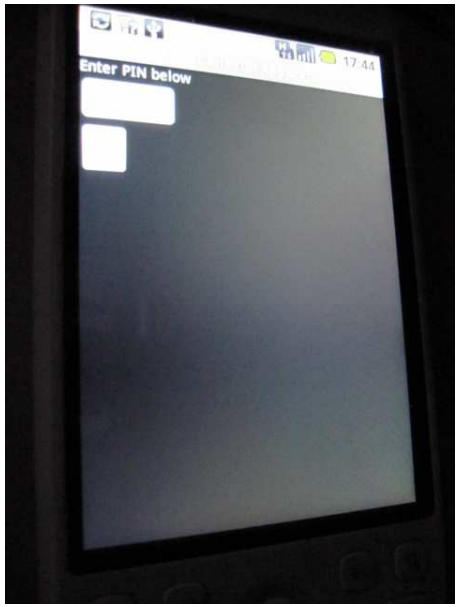


Figure: Generated patient application running on an *HTC Magic* phone

An eService Scenario... in the flesh

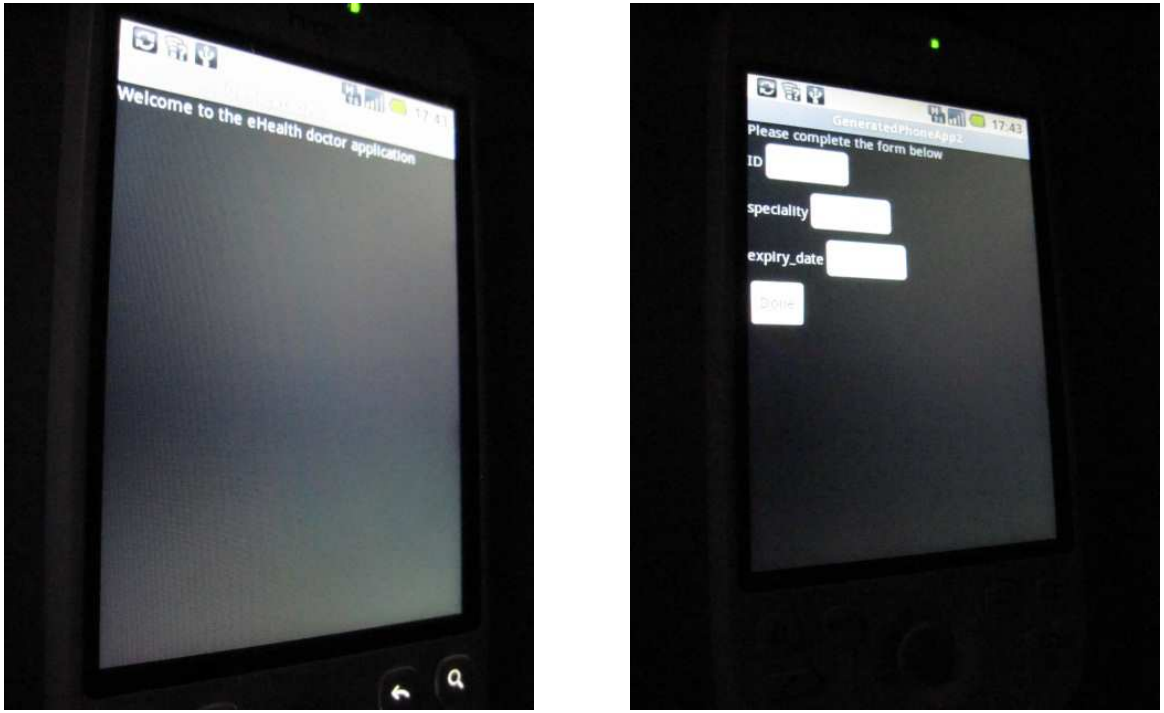


Figure: Generated doctor application running on an *HTC Magic* phone

An eService Scenario... in the flesh

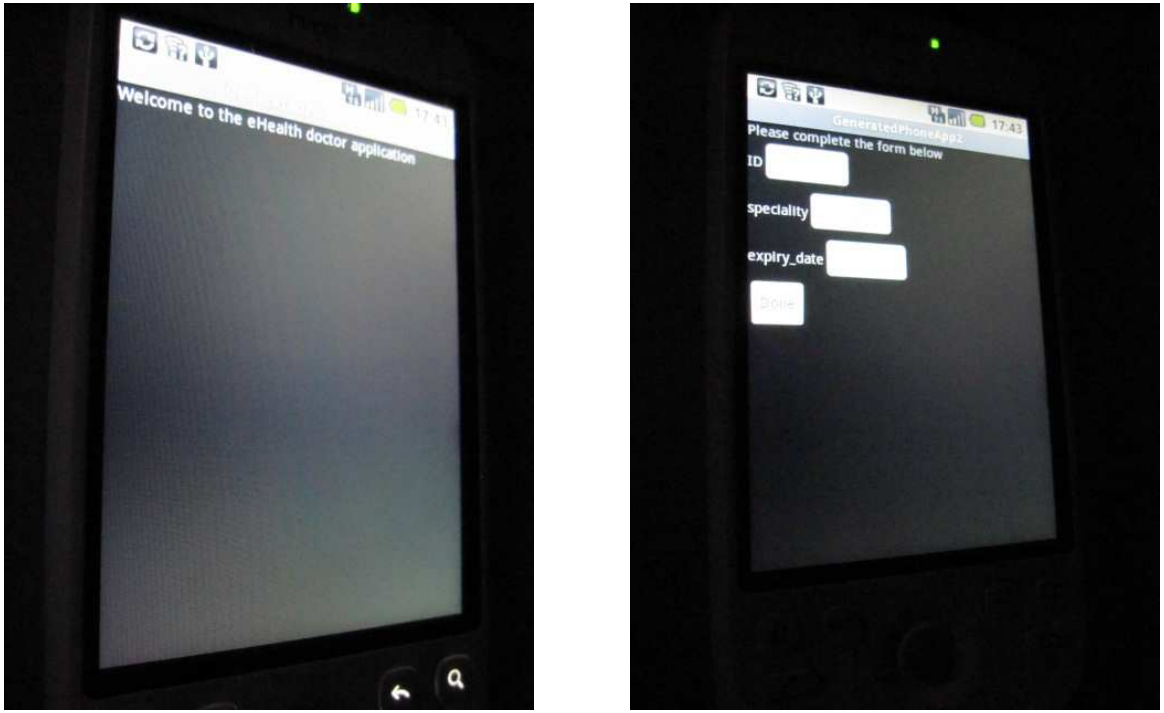


Figure: Generated doctor application running on an *HTC Magic* phone

And now, a demonstration... (3 videos, URLs below)

http://msdl.cs.mcgill.ca/people/raphael/files/adapid09-vid1.avi

http://msdl.cs.mcgill.ca/people/raphael/files/adapid09-vid2.avi

http://msdl.cs.mcgill.ca/people/raphael/files/adapid09-vid3.avi

Future Work

- Extending the modeling primitives
  $\longrightarrow$ wider variety of privacy-preserving applications;
- Extending the model transformations
  $\longrightarrow$ wider variety of target platforms and more polished applications;
- Moving towards the Distrinet Framework
  $\longrightarrow$ standardized and more robust back-end;
- Model analyses and simulation
  $\longrightarrow$ privacy property verification (e.g. linkability);

Conclusions

- Modeling
  → is useful for documentation and communication;
  → is amenable to analyses and proofs;

- Domain-Specific Modeling
  → hides conceptual gaps;
  → leverages expertise of programmers and non-programmers;
  → considerably reduces development times;
  → hides commonalities between eServices;

Questions?

Thank you!

References

- A privacy-preserving ehealth protocol compliant with the belgian healthcare system. Bart De Decker, Mohamed Layouni, Hans Vangheluwe, and Kristof Verslype. *In Proc. of the fifth European PKI Workshop (EuroPKI08), Lecture Notes in Computer Science*, vol. 5057, Springer, 2008, pp. 118-133.

- Privacy-Preserving Telemonitoring for eHealth. Mohamed Layouni, Kristof Verslype, Mehmet Tahir Sandkkaya, Bart De Decker, and Hans Vangheluwe. *In Proc. of the 23rd Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec09), Lecture Notes in Computer Science*, vol. 5645, Springer, 2009, pp. 95-110.