Automatic Analysis of Security Protocols

Roberto Carbone

http://st.fbk.eu

Security & Trust Research Unit

Center for Information Technologies

Bruno Kessler Foundation

Motivations

- Security protocols and services are key to securing the ever-growing ecosystem of online applications (web, mobile, ...)
- But security solutions are notoriously difficult to get right. Many security-critical protocols and services have been designed and developed only to be found flawed years after their deployment.
- Due to the complex and unexpected interleaving of the protocols and services as well as to the possible interference of malicious agents.
- Very difficult to spot by traditional verification techniques (e.g., manual inspection and testing)
- Security-critical systems are a natural target for formal method techniques.

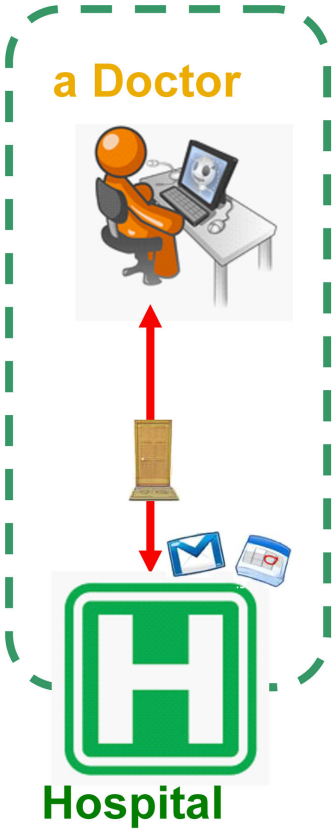
Motivations

- Security protocols and services are key to securing the ever-growing ecosystem of online applications (web, mobile, ...)
- But security solutions are notoriously difficult to get right. Many security-critical protocols and services have been designed and developed only to be found flawed years after their deployment.
- Due to the complex and unexpected interleaving of the protocols and services as well as to the possible interference of malicious agents.
- Very difficult to spot by traditional verification techniques (e.g., manual inspection and testing)
- Security-critical systems are a natural target for formal method techniques.

Outline

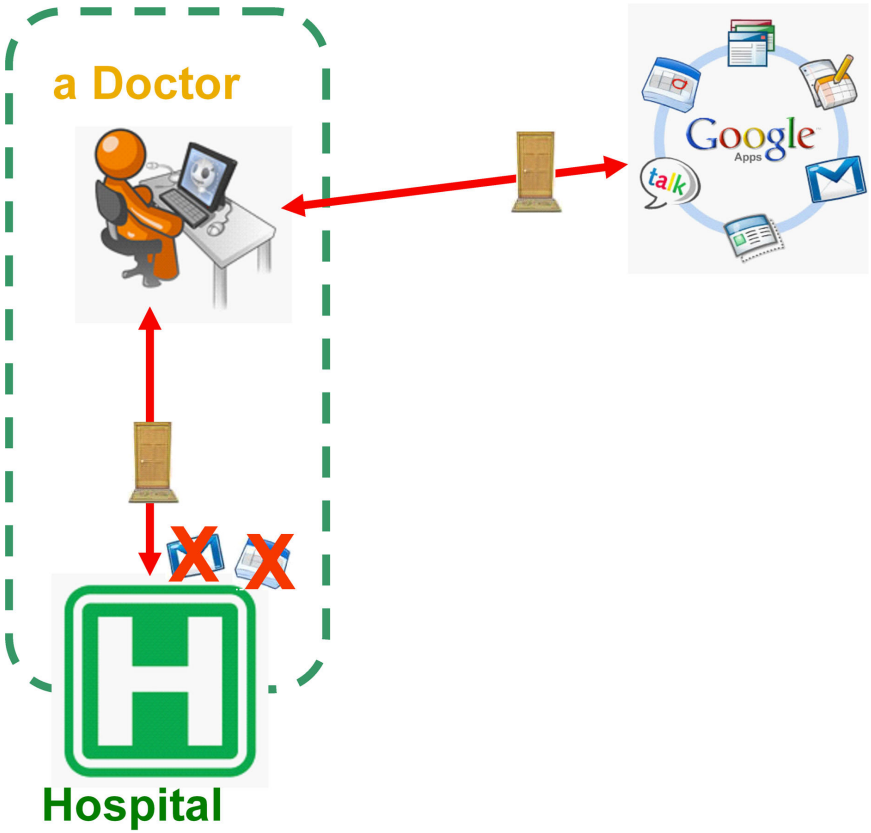
Browser-based Single Sign-On: Use Case

- Hospital outsources basic IT services $\Rightarrow$ *Google Apps*
- Identity management $\Rightarrow$ *SAML-based Single Sign On*



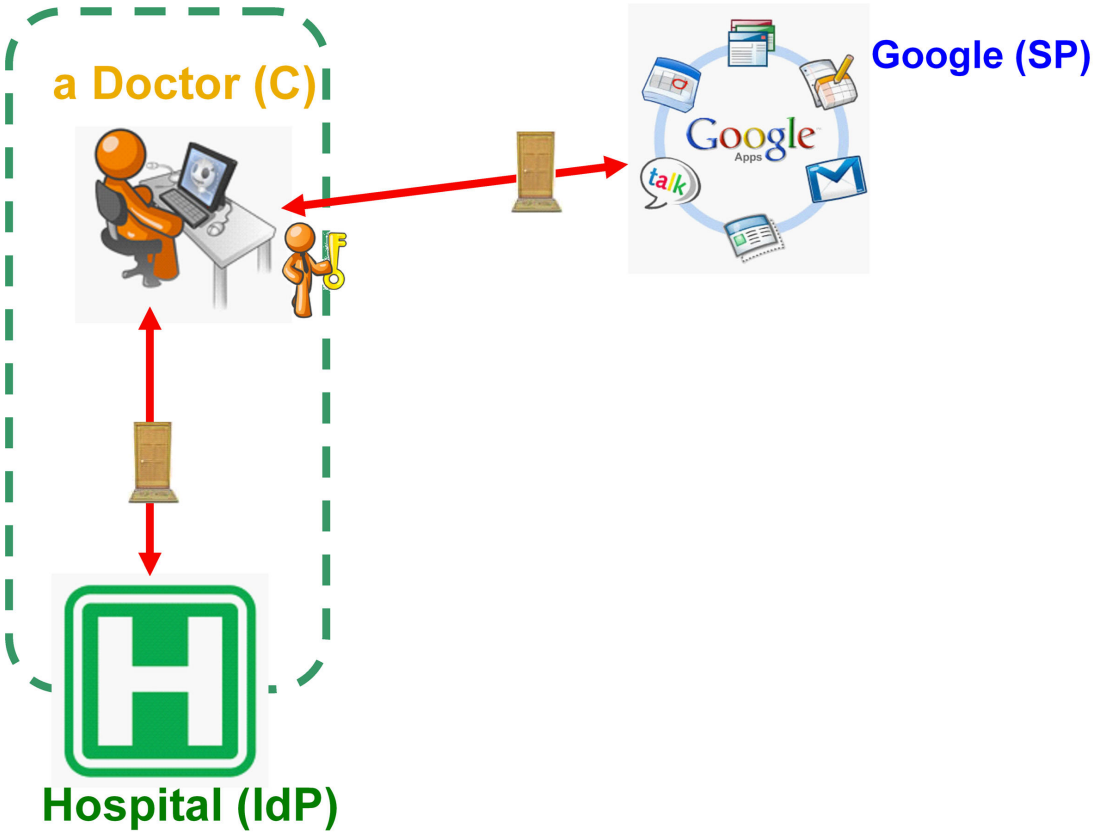
Browser-based Single Sign-On: Use Case

- Hospital outsources basic IT services ⇒ *Google Apps*
- Identity management ⇒ *SAML-based Single Sign On*



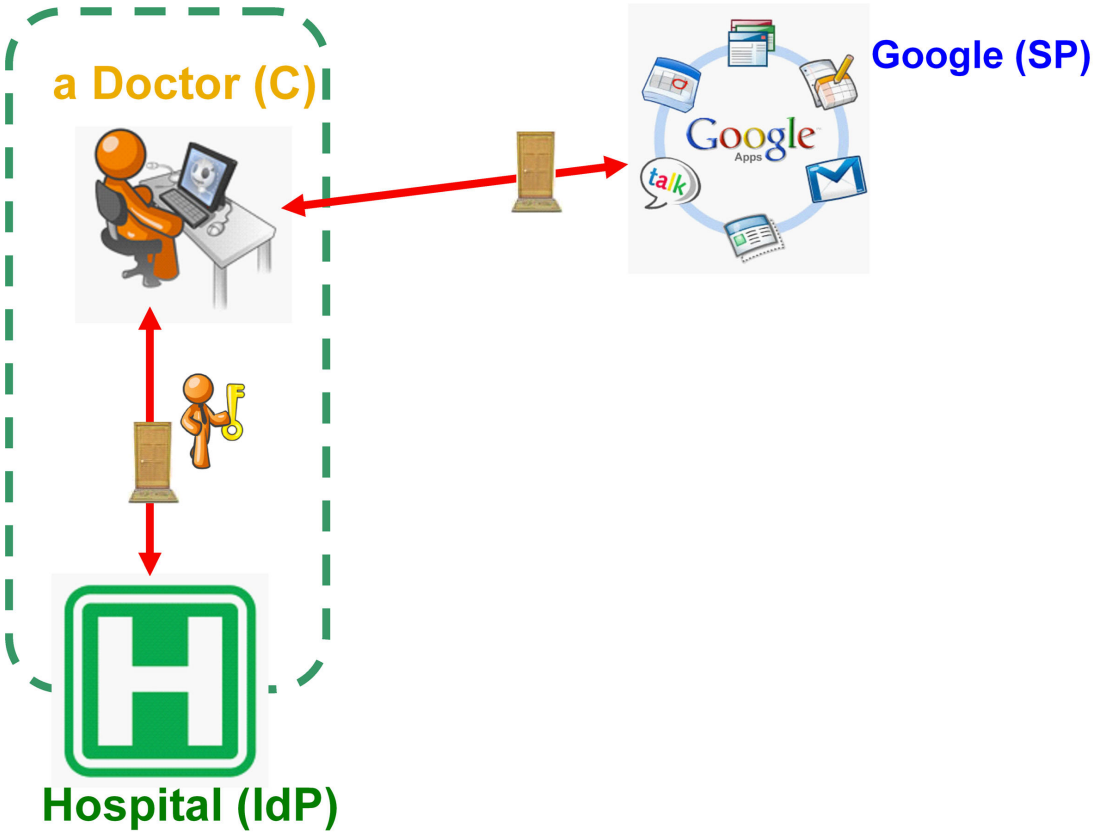
Browser-based Single Sign-On: Use Case

- Hospital outsources basic IT services $\Rightarrow$ *Google Apps*
- Identity management $\Rightarrow$ *SAML-based Single Sign On*



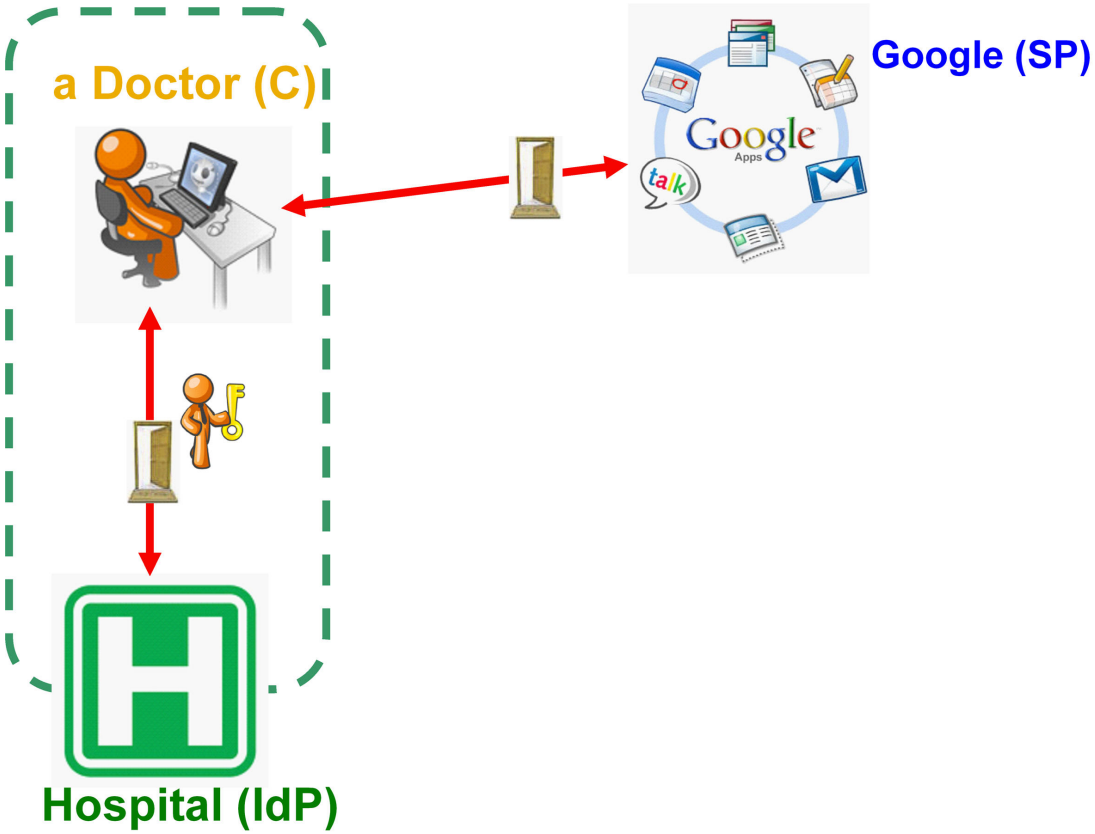
Browser-based Single Sign-On: Use Case

- Hospital outsources basic IT services $\Rightarrow$ *Google Apps*
- Identity management $\Rightarrow$ *SAML-based Single Sign On*



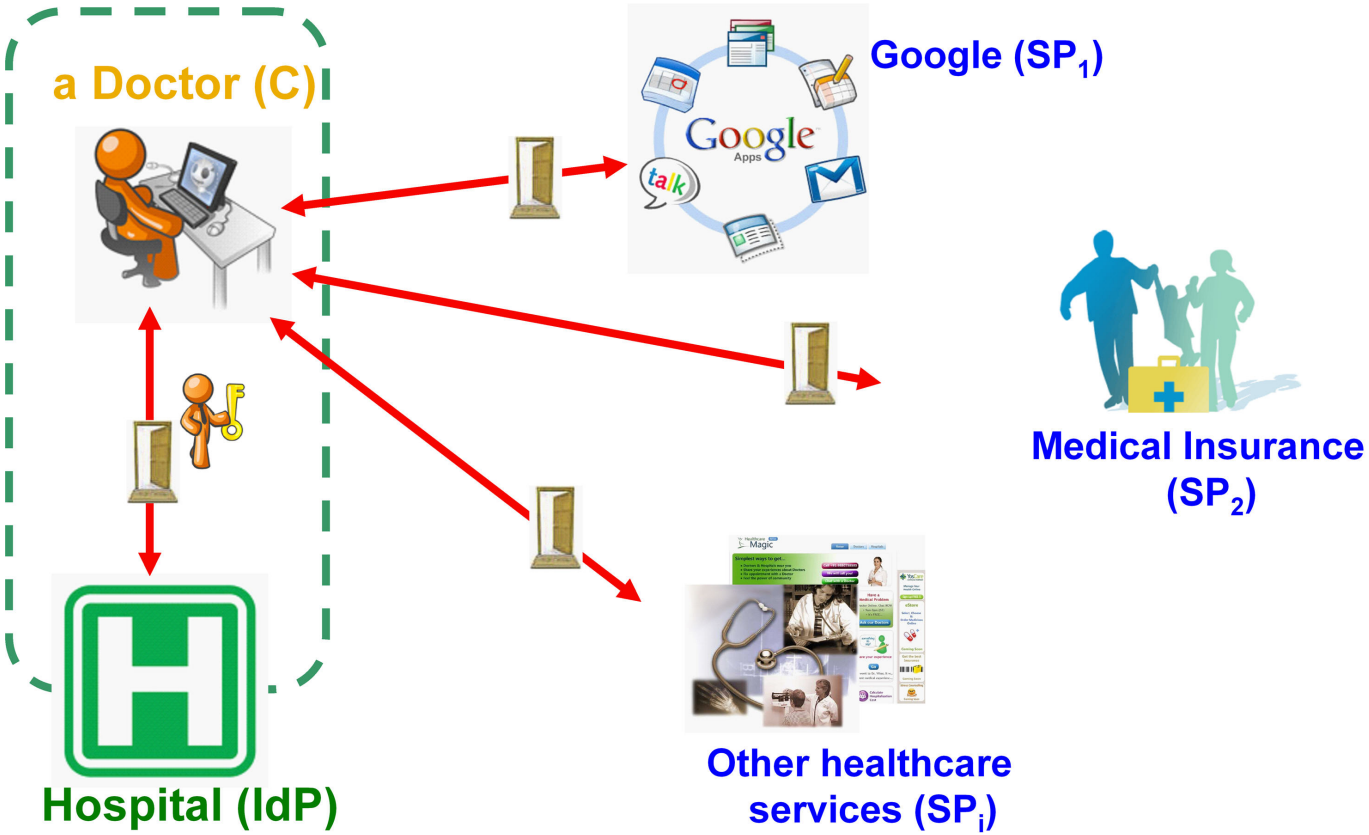
Browser-based Single Sign-On: Use Case

- Hospital outsources basic IT services $\Rightarrow$ *Google Apps*
- Identity management $\Rightarrow$ *SAML-based Single Sign On*

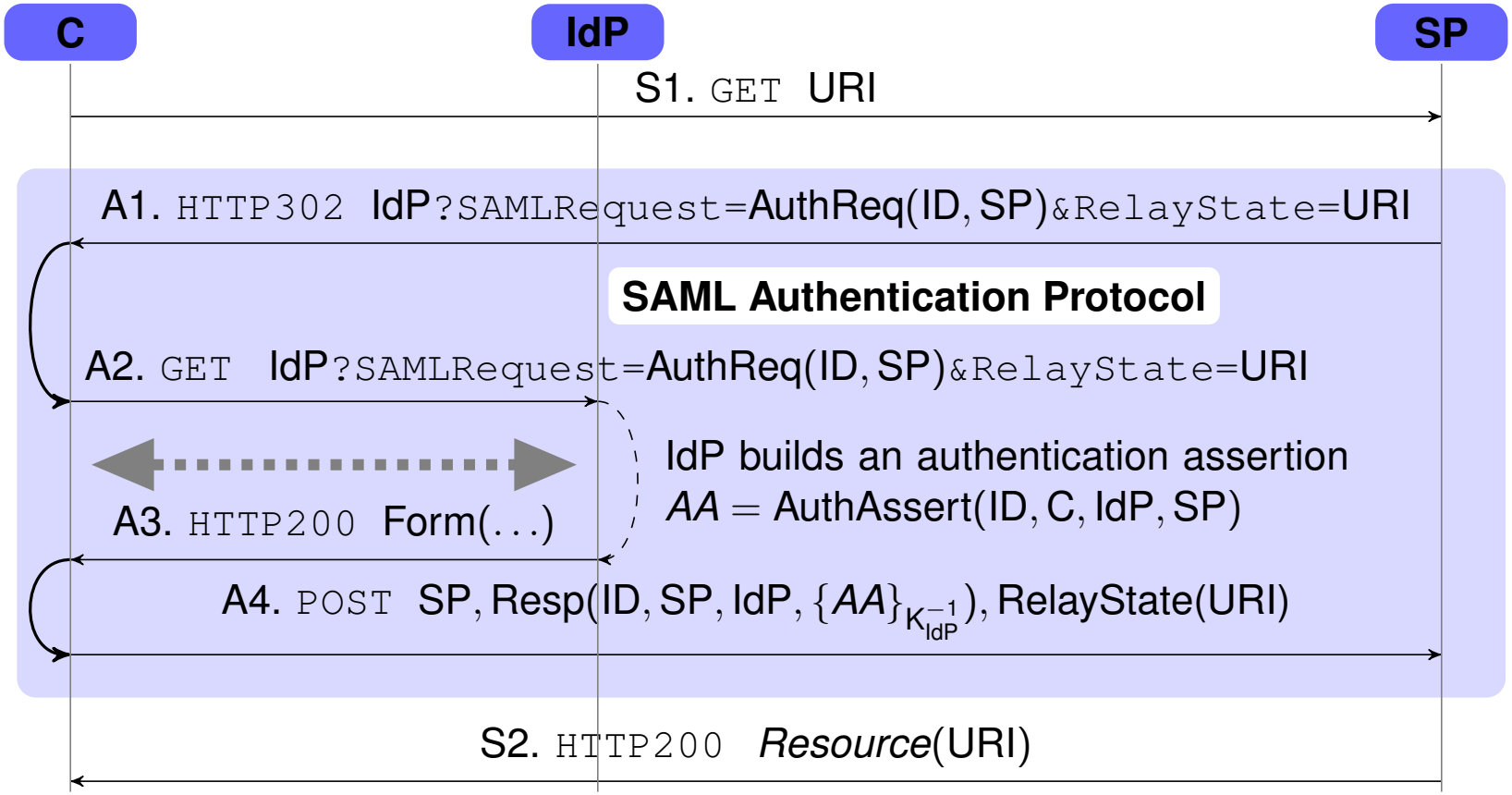


Browser-based Single Sign-On: Use Case

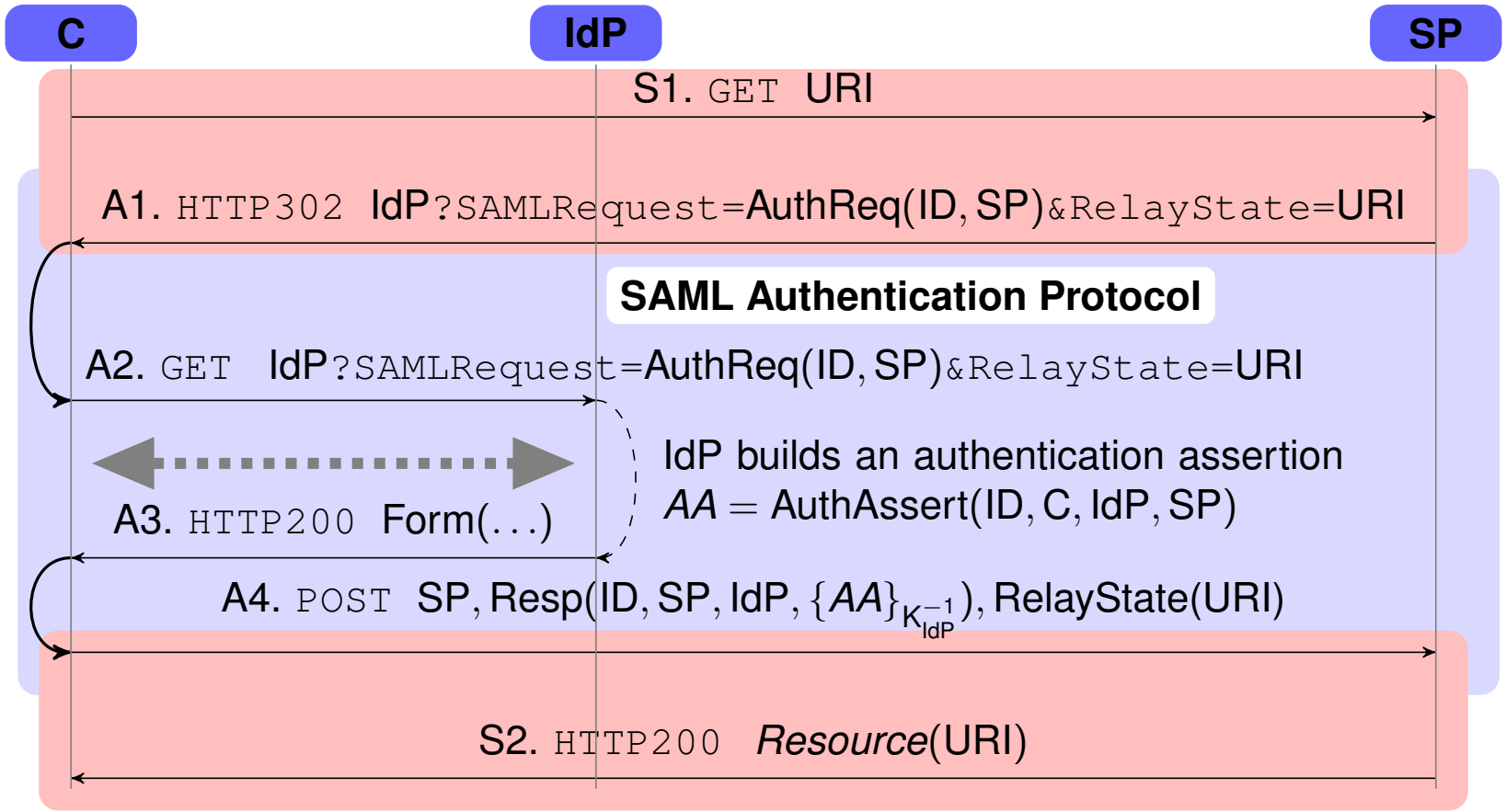
- Hospital outsources basic IT services $\Rightarrow$ *Google Apps*
- Identity management $\Rightarrow$ *SAML-based Single Sign On*



The SAML 2.0 Web Browser SSO Profile (SAML SSO)



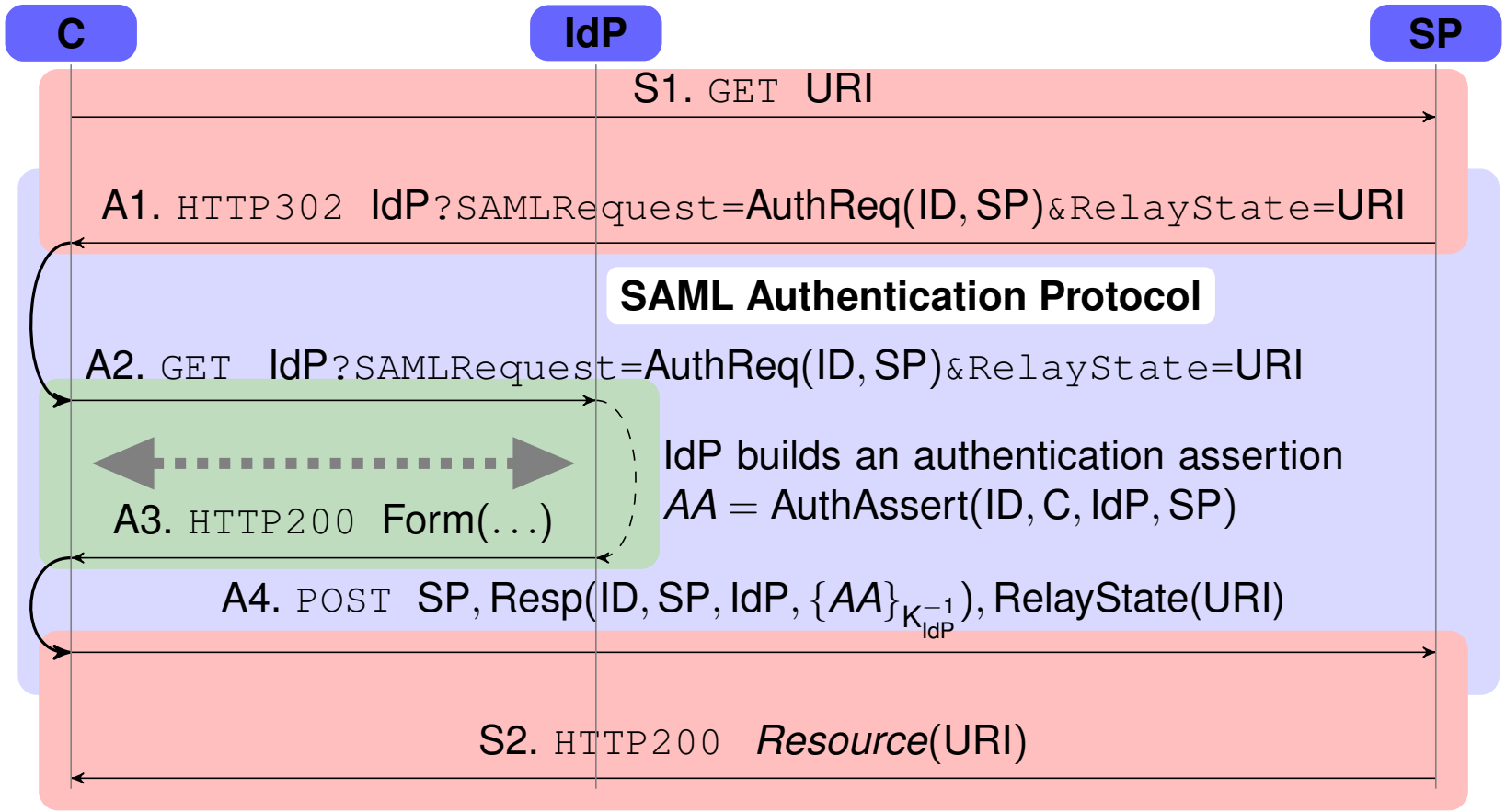
The SAML 2.0 Web Browser SSO Profile (SAML SSO)



Assumption on Transport Protocols (TP1)

Communication between C and SP is carried over a unilateral SSL/TLS channel.

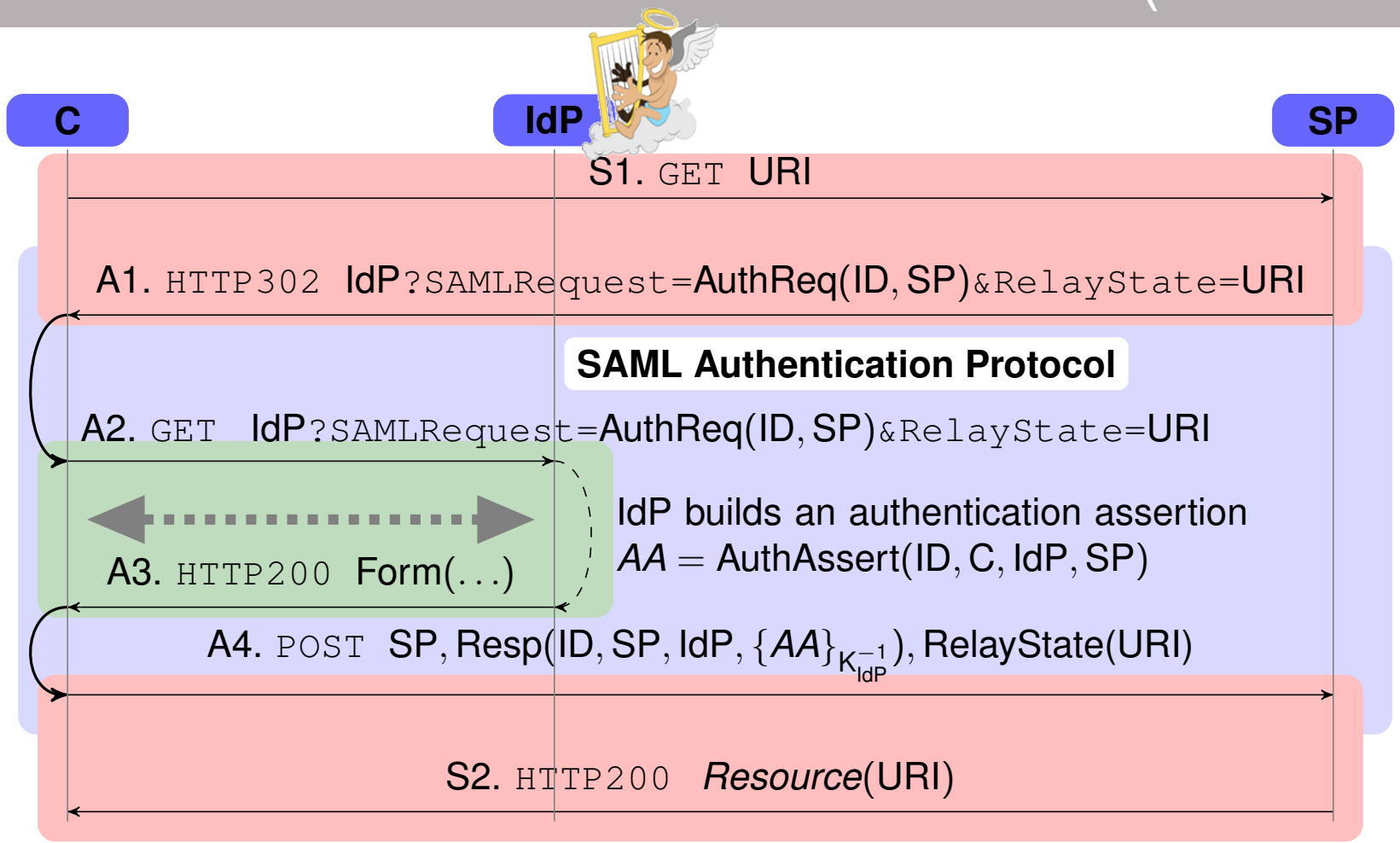
The SAML 2.0 Web Browser SSO Profile (SAML SSO)



Assumption on Transport Protocols (TP2)

Communication between C and IdP is carried over a unilateral SSL/TLS channel that becomes bilateral once C authenticates on IdP.

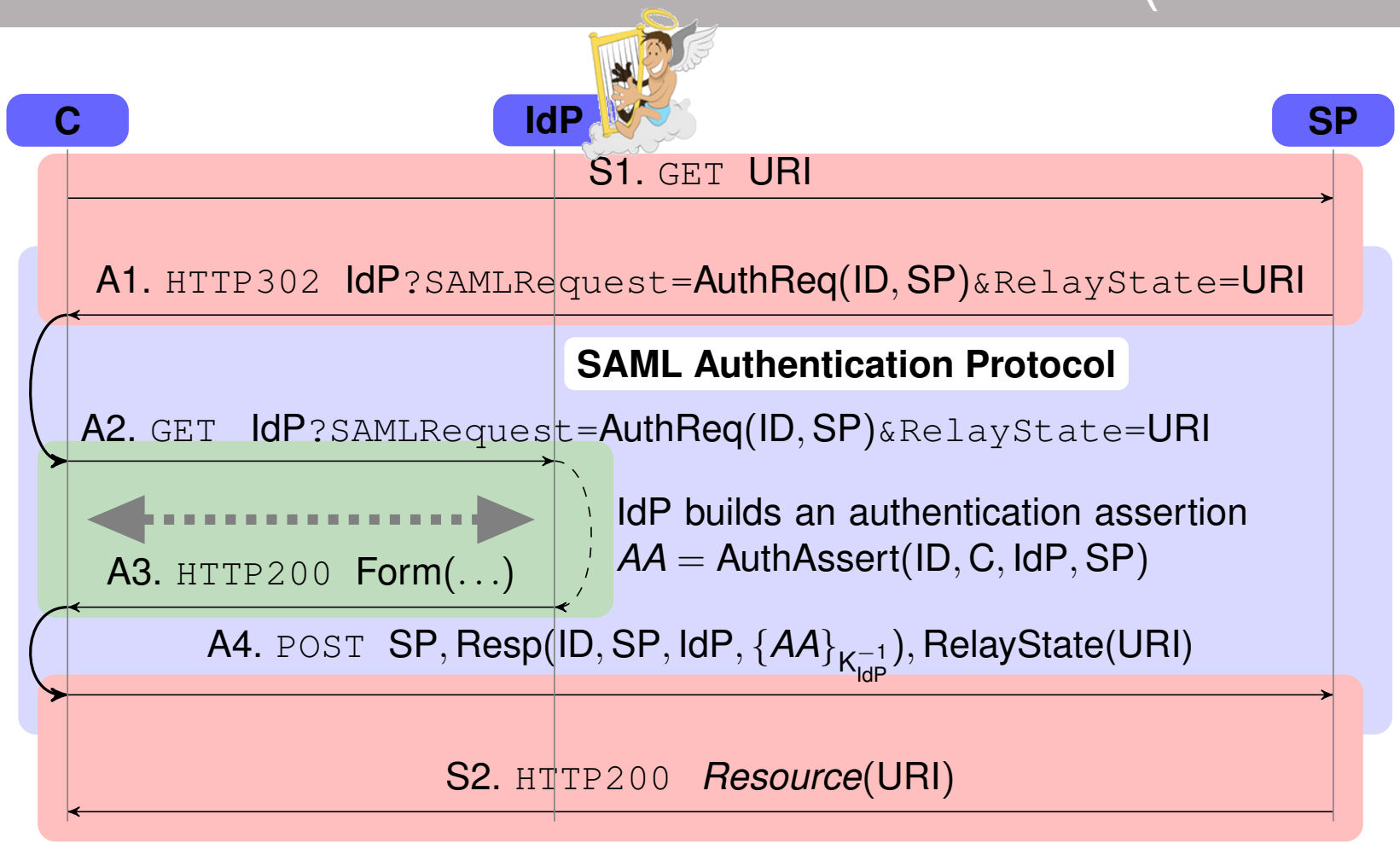
The SAML 2.0 Web Browser SSO Profile (SAML SSO)



Trust Assumption (TA1)

IdP is not compromised, i.e. it is not under the control of an intruder and it abides by the rules of the protocol.

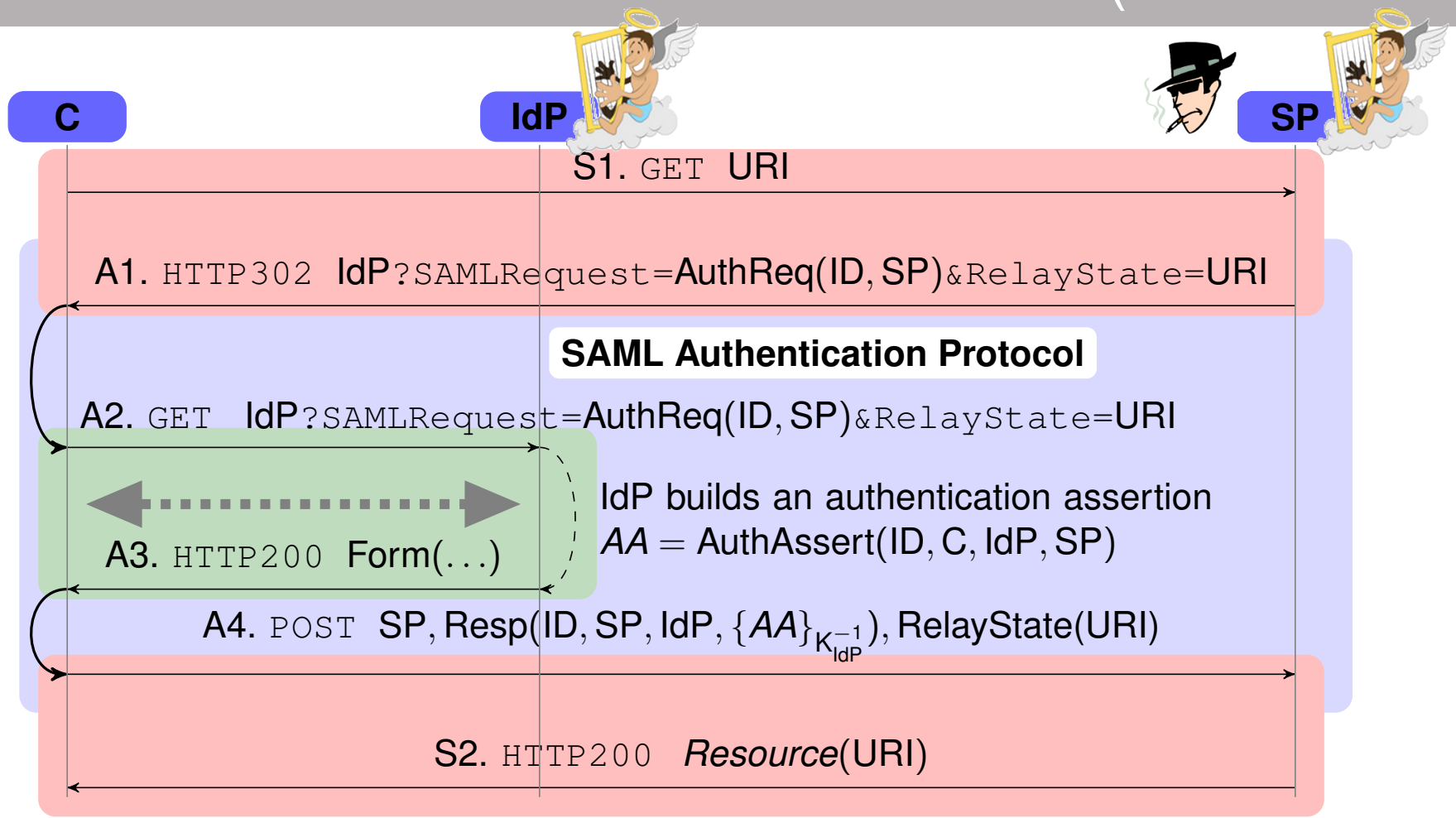
The SAML 2.0 Web Browser SSO Profile (SAML SSO)



Trust Assumption (TA2)

IdP is trusted by SP to generate authentication assertions about C.

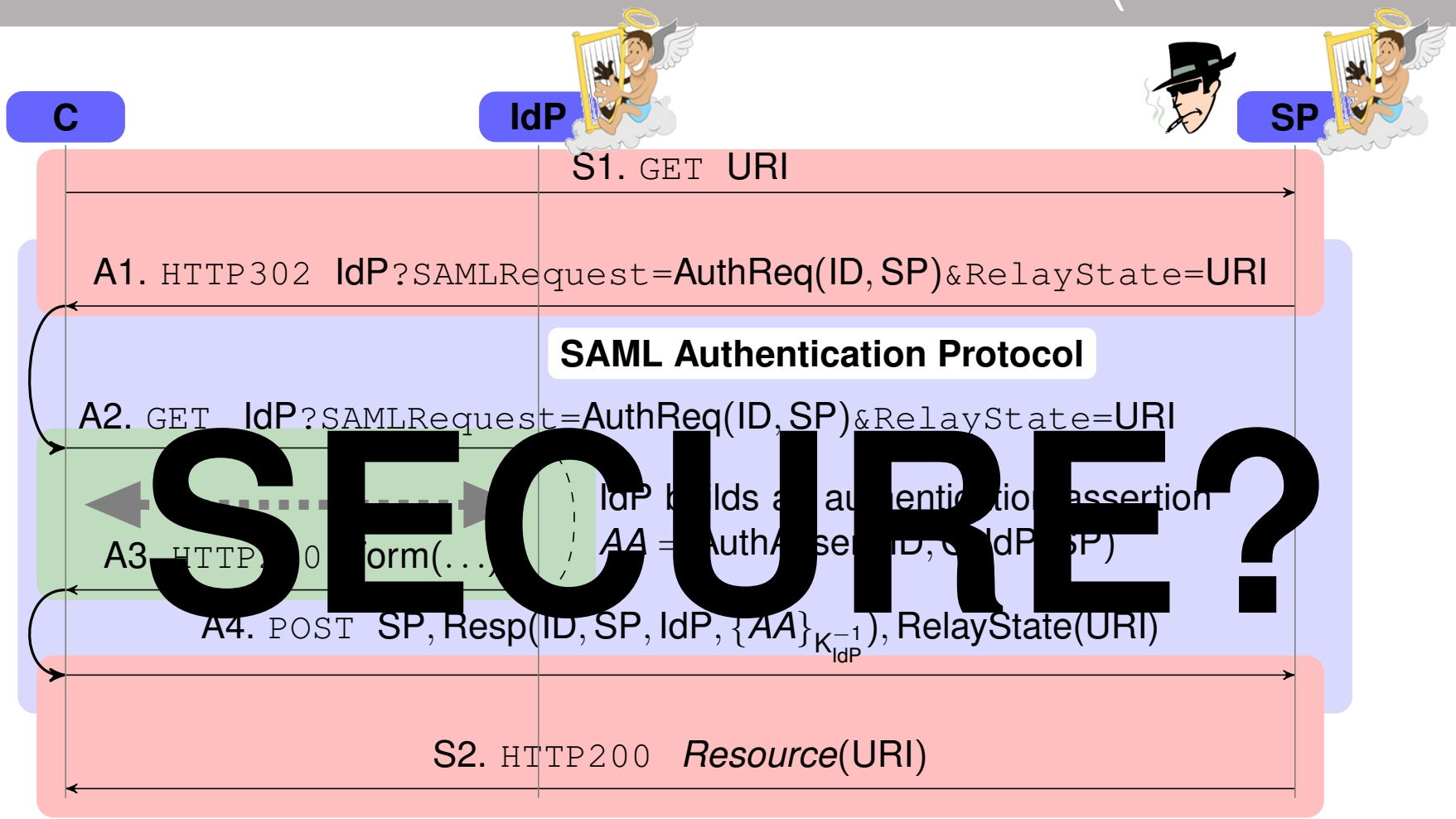
The SAML 2.0 Web Browser SSO Profile (SAML SSO)



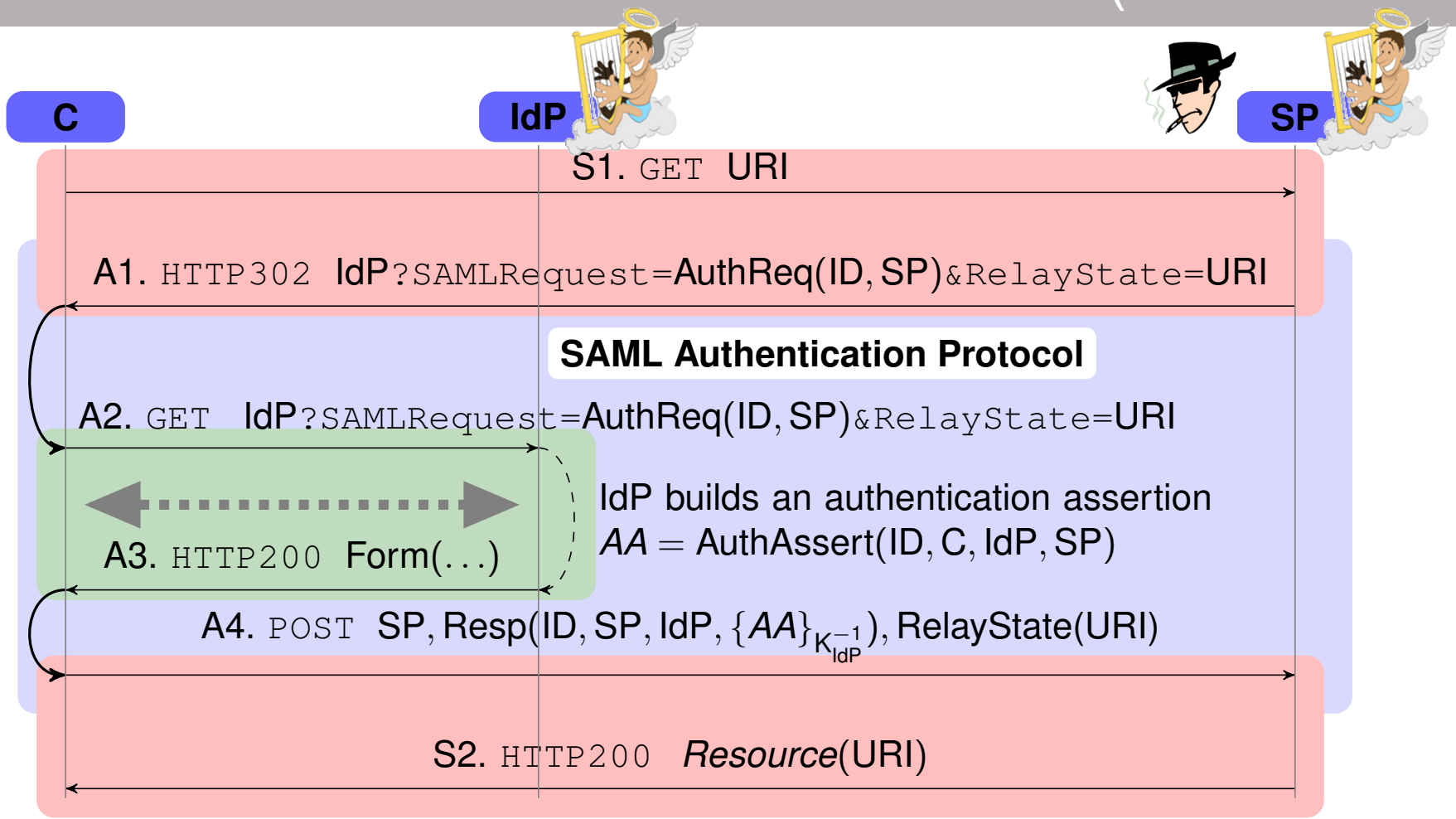
Important

We do not assume that all SPs whom C may play the protocol with are uncompromised.

The SAML 2.0 Web Browser SSO Profile (SAML SSO)



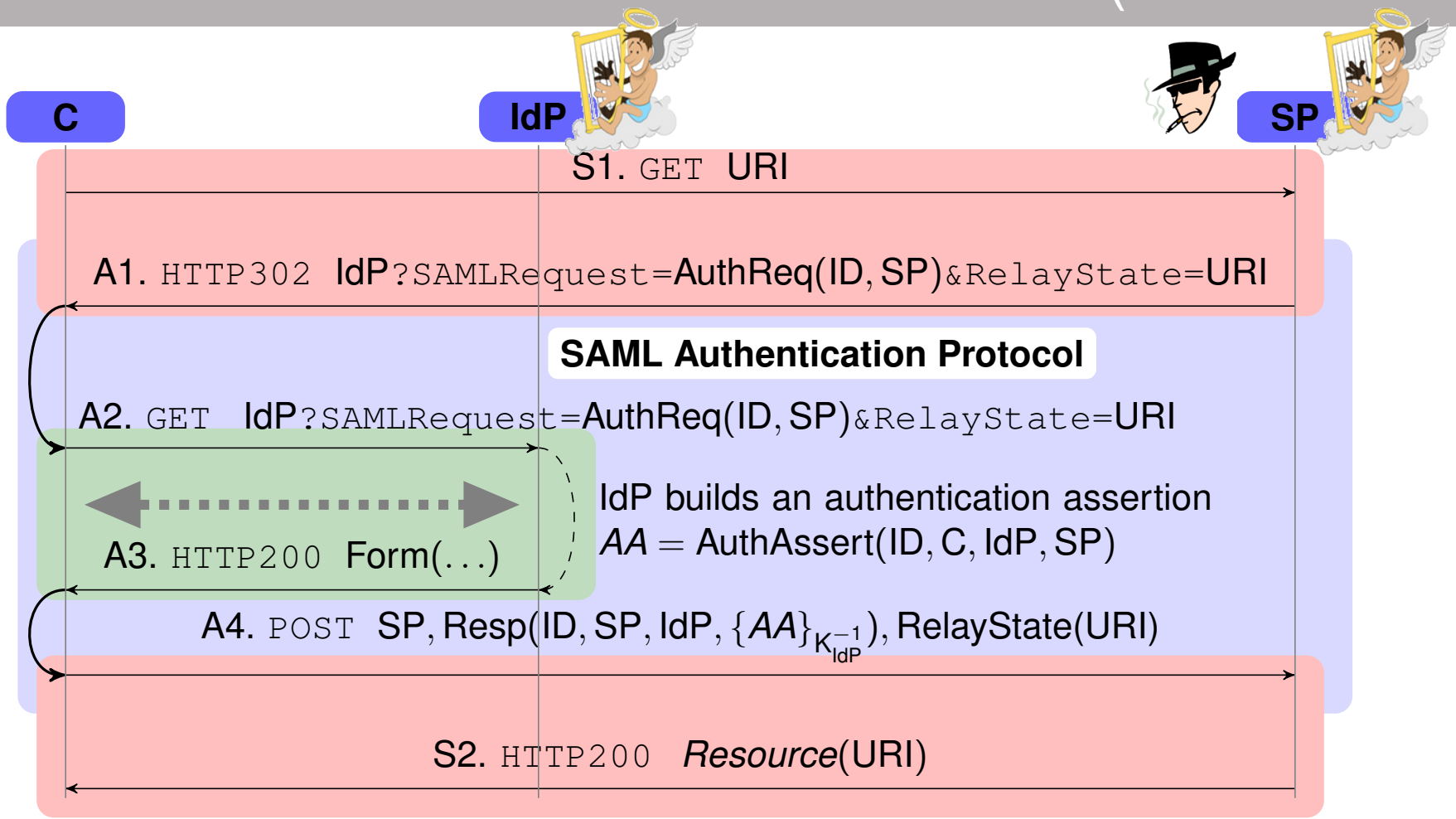
The SAML 2.0 Web Browser SSO Profile (SAML SSO)



Security Goal (SAML SSO)

SP and C mutually authenticate and agree on URI

The SAML 2.0 Web Browser SSO Profile (SAML SSO)



Security Goal (SAML Authentication Prococol)

SP authenticates C

Outline

SATMC: a Bounded Model Checker for Security Protocols

- SATMC tackles problems of the form:

$$(P \| I), C \models G$$

where
  - P: transition system modeling honest participants.
  - I: transition system modeling DY intruder.
  - C: LTL formula constraining the behaviours of DY intruder on the communication channels.
  - G: LTL formula encoding the expected security properties.
- Successful combination of
  - SAT-reduction techniques developed for AI-planning
  - Bounded model-checking techniques developed for reactive systems.

A. Armando, R. Carbone, L. Compagna. “SATMC: a SAT-based Model Checker for Security-critical Systems”, In Proc. 20th international Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS’14), 2014.

Outline

The “SAML-based” SSO for Google Apps

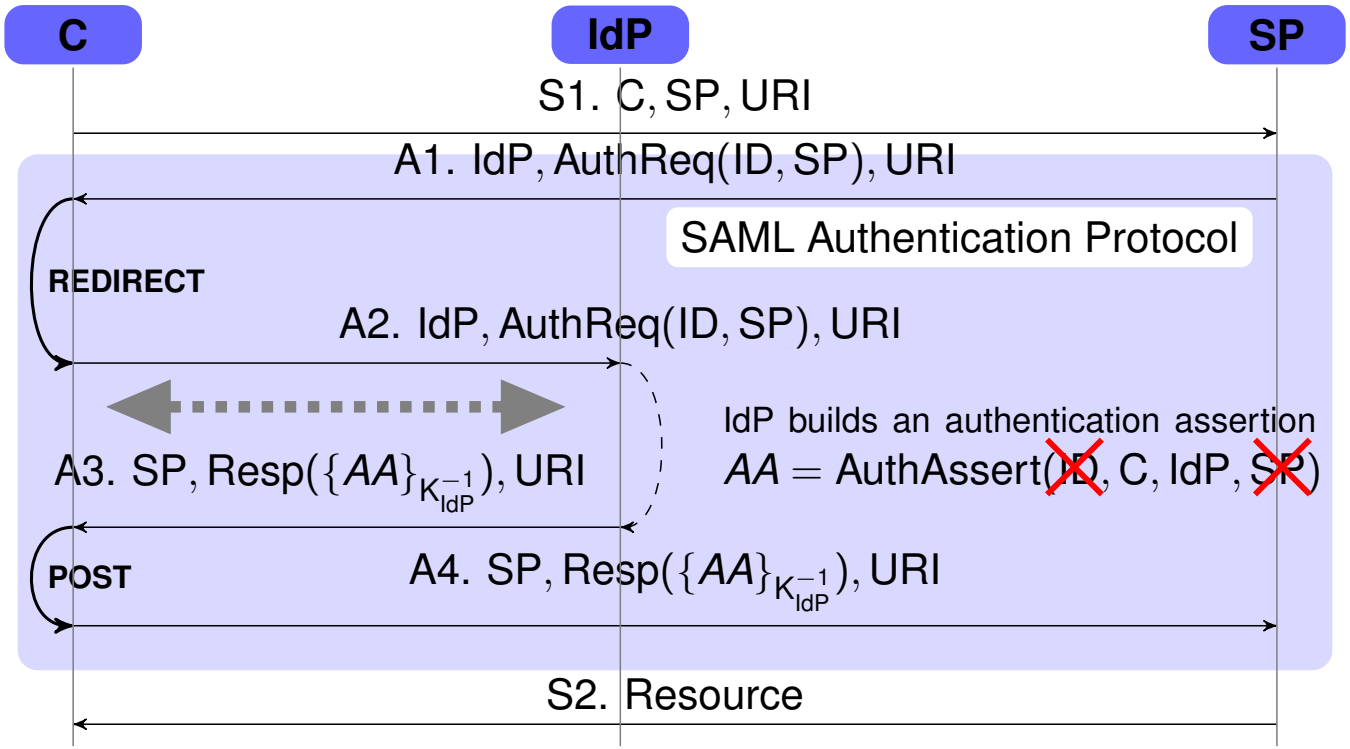
Same as the SAML 2.0 Web Browser SSO Profile except for seemingly minor simplifications:

- ID and SP are not included in the authentication assertion.

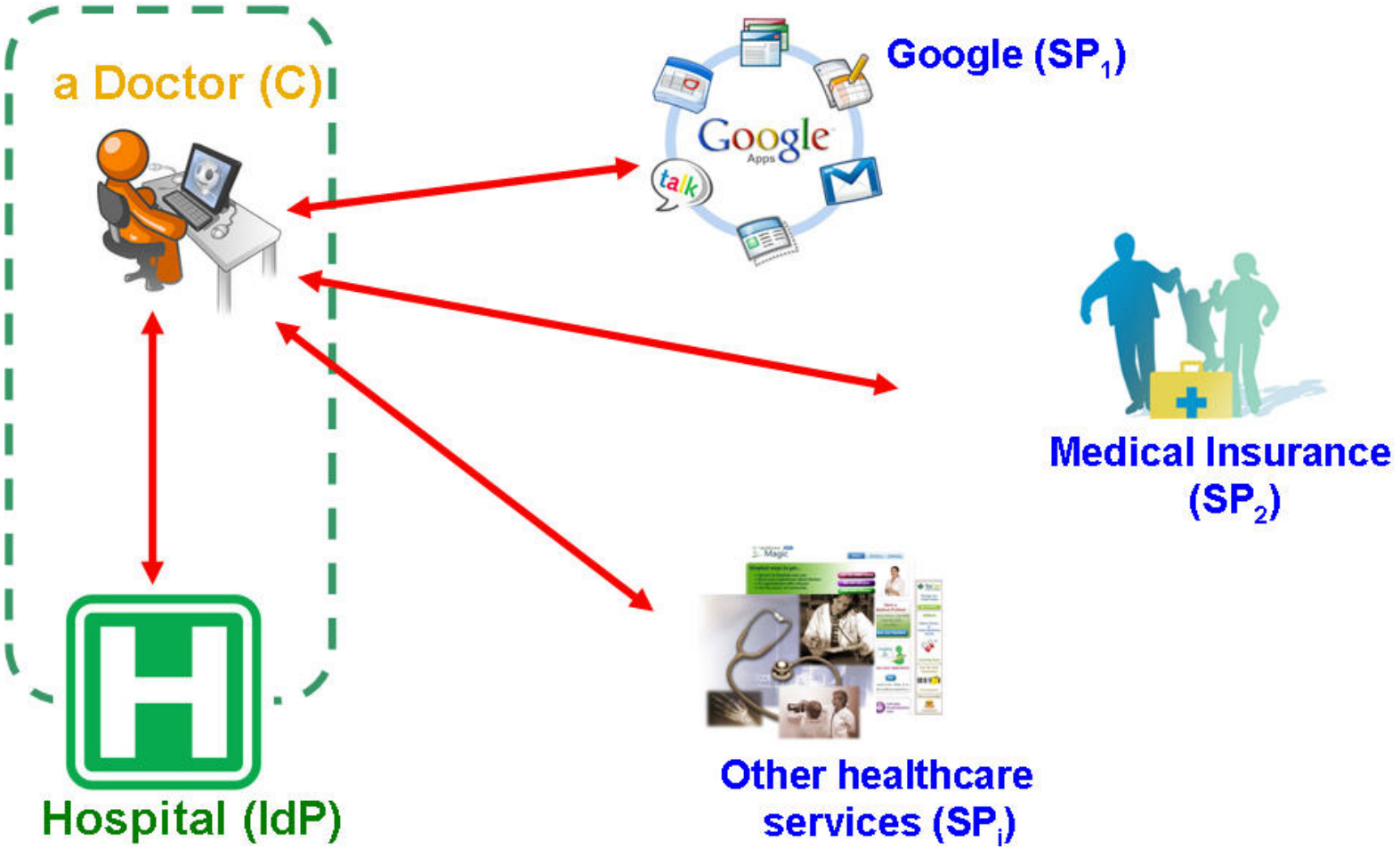
The “SAML-based” SSO for Google Apps

Same as the SAML 2.0 Web Browser SSO Profile except for seemingly minor simplifications:

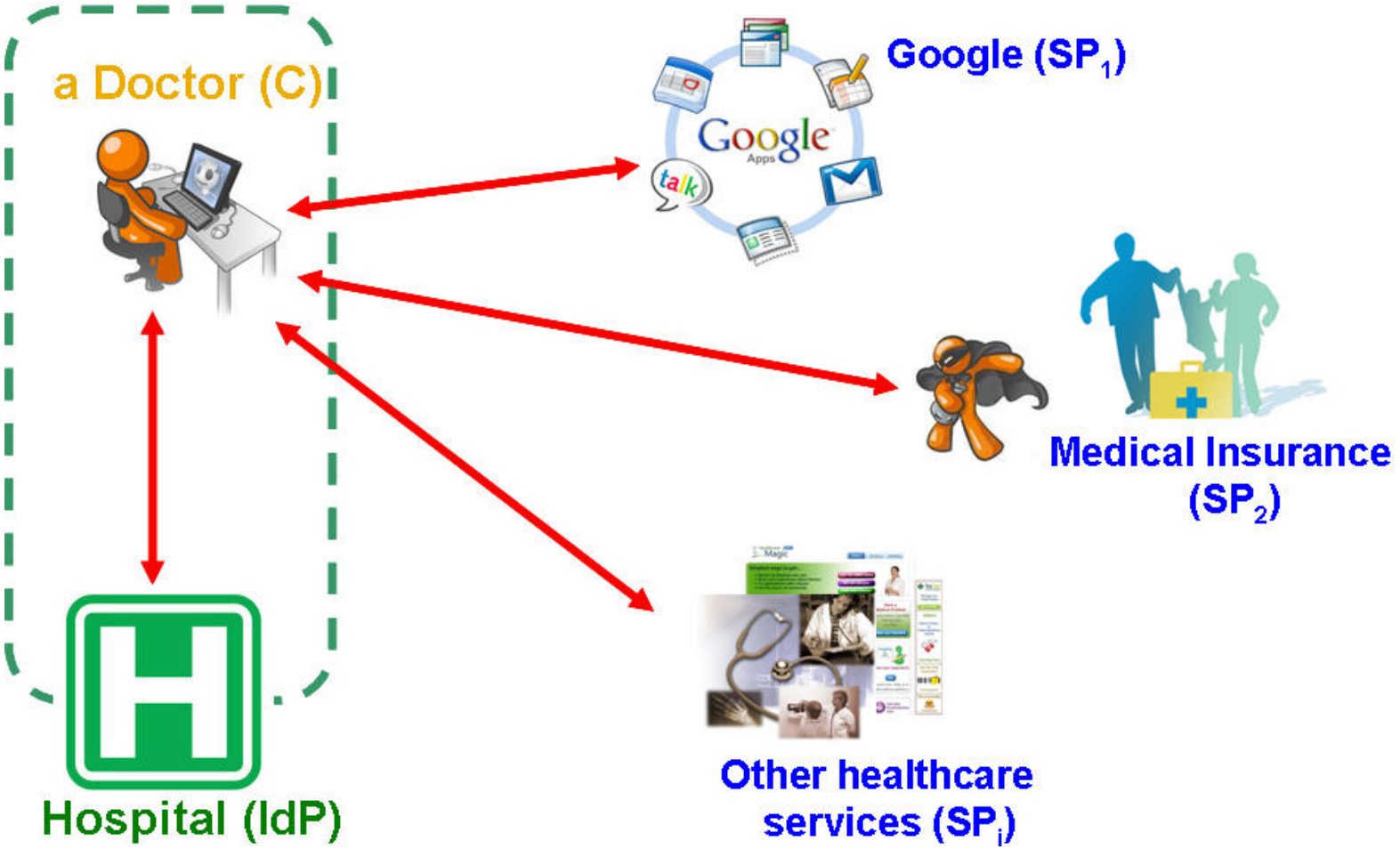
- ID and SP are not included in the authentication assertion.



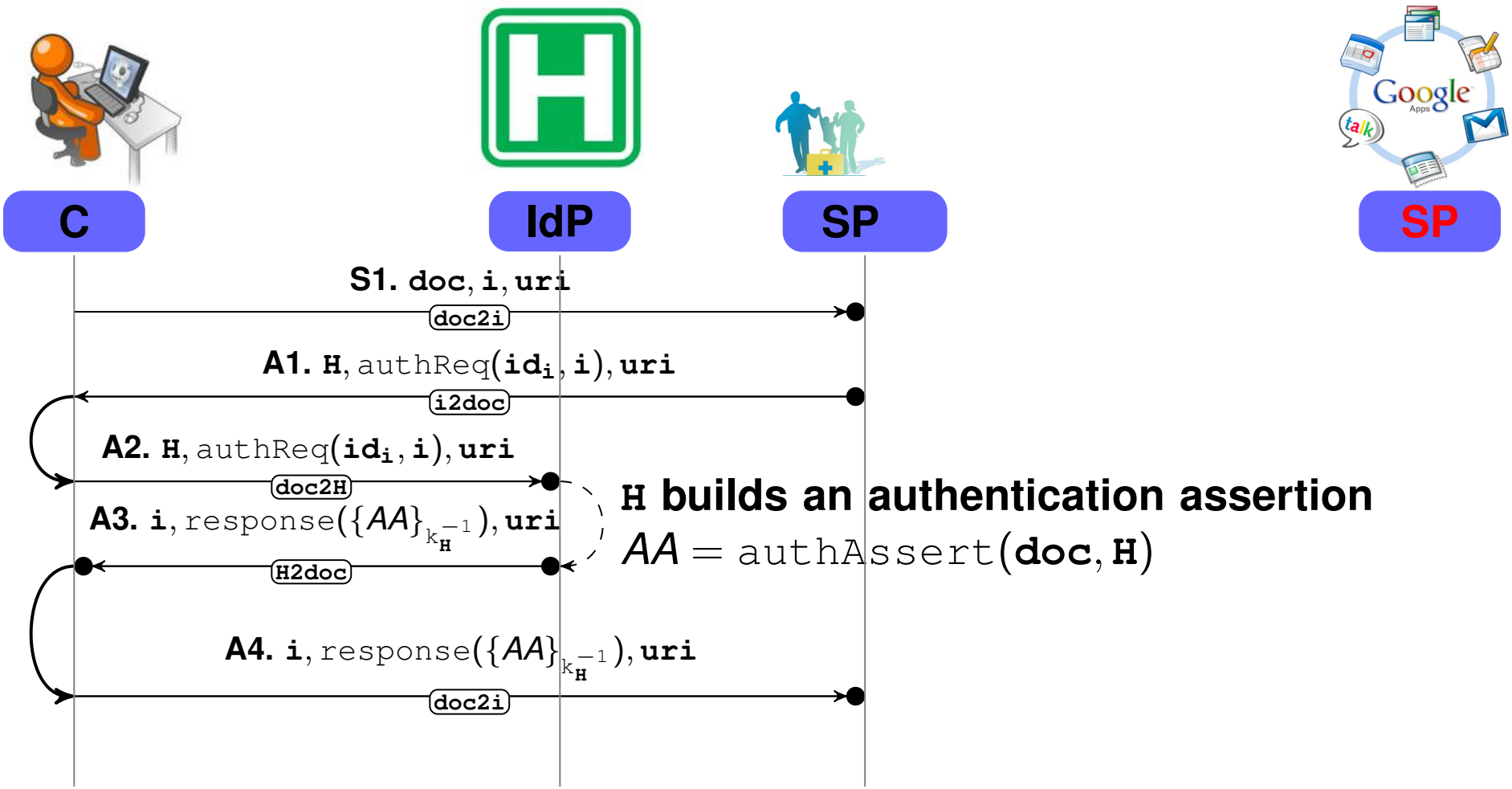
Use Case Analysis



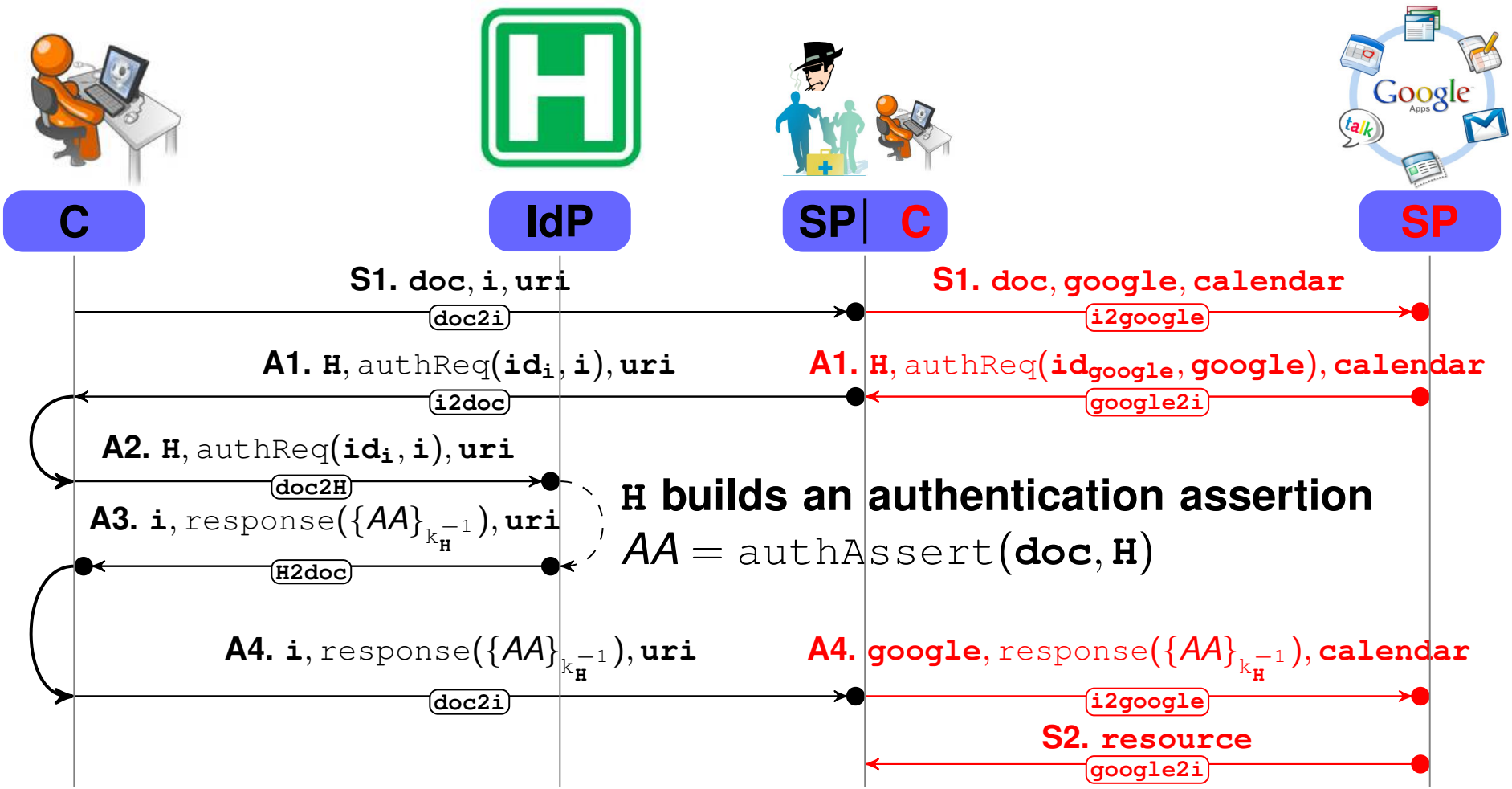
Use Case Analysis



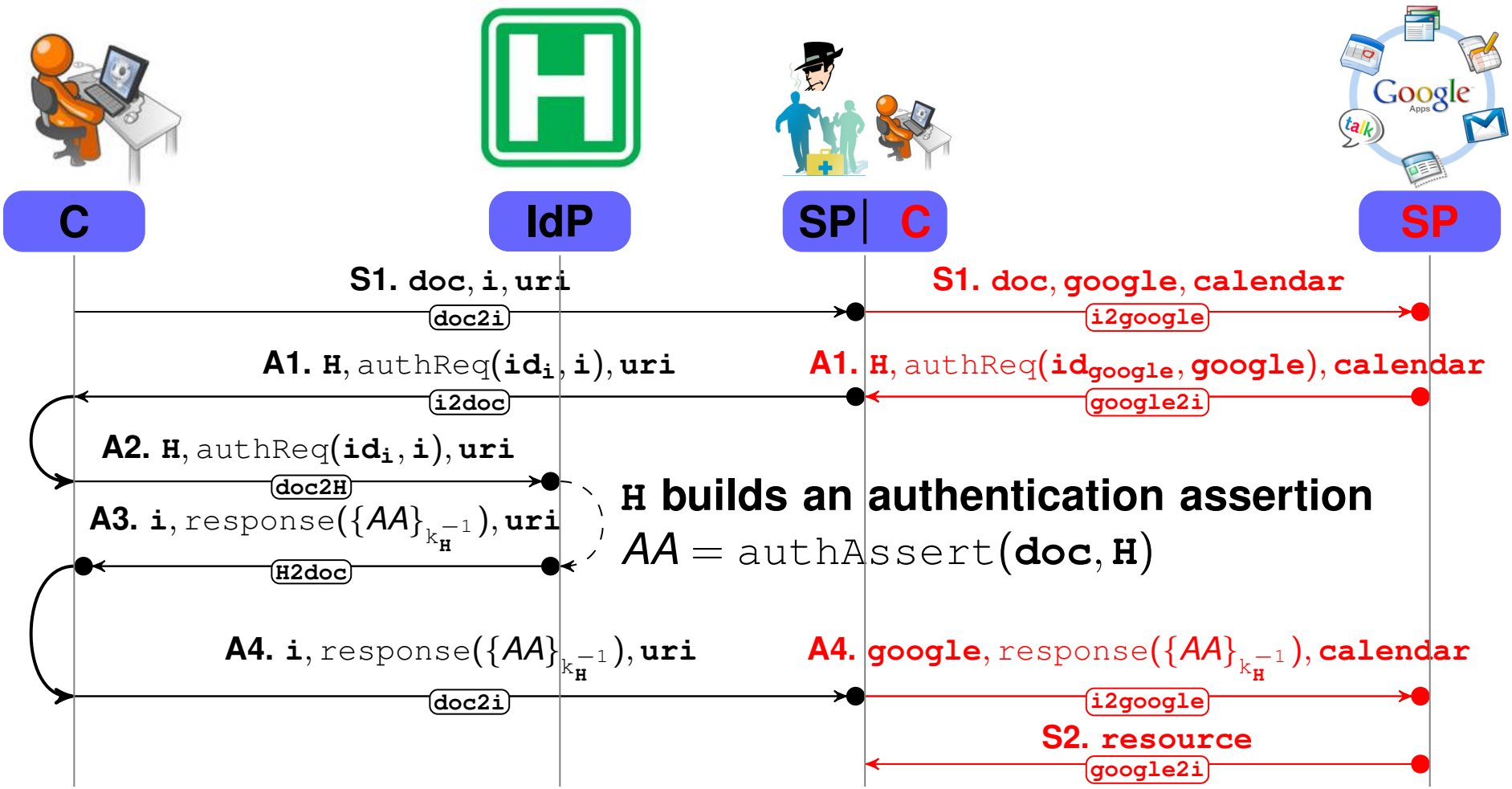
Attack on the SAML-based SSO for Google Apps



Attack on the SAML-based SSO for Google Apps



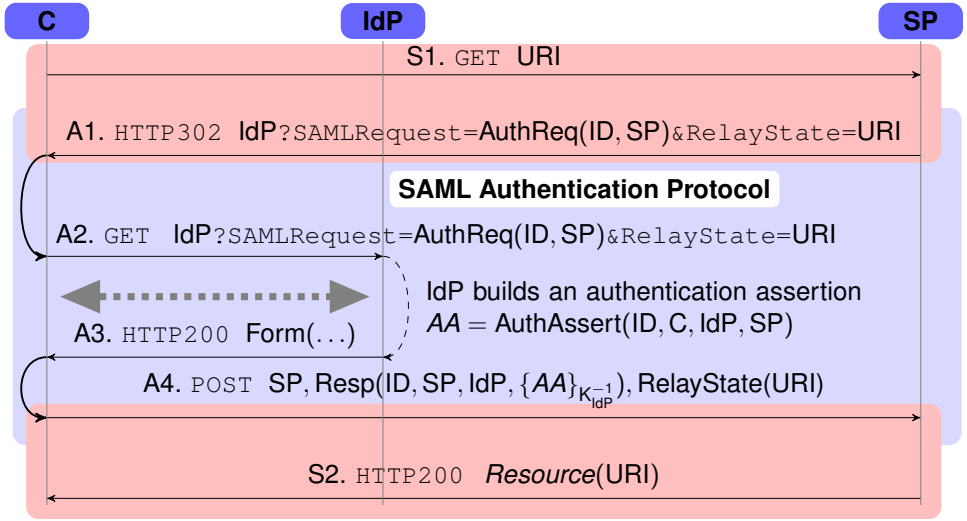
Attack on the SAML-based SSO for Google Apps



A. Armando, R. Carbone, L. Compagna, J. Cuéllar and L. Tobarra. **Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-based Single Sign-On for Google Apps.** In the Proceedings of the 6th ACM Workshop on Formal Methods in Security Engineering (FMSE 2008), 2008, Virginia, USA.

Outline

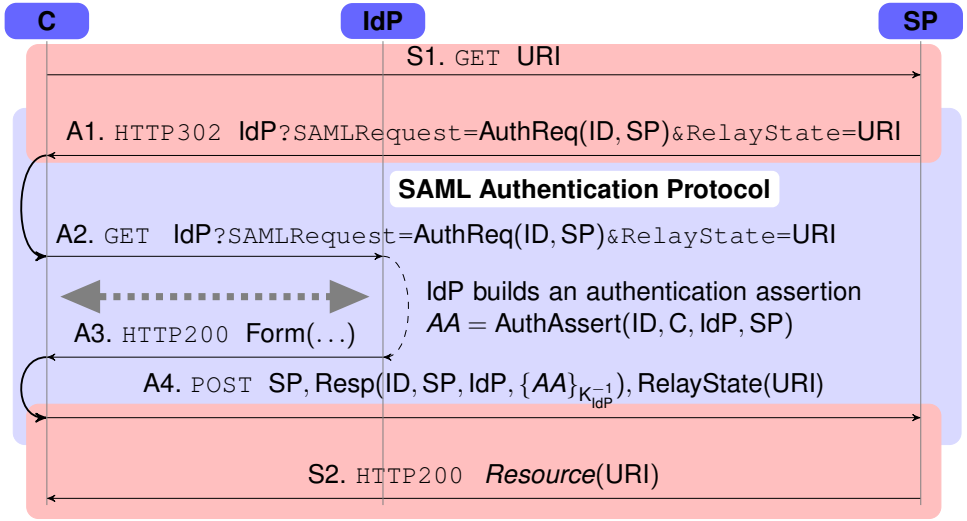
The SAML 2.0 Web Browser SSO Profile (SAML SSO)



Assumption on Transport Protocols (TP1)

Communication between C and SP is carried over a unilateral SSL/TLS channel.

The SAML 2.0 Web Browser SSO Profile (SAML SSO)



Assumption on Transport Protocols (TP1)

Communication between C and SP is carried over a unilateral SSL/TLS channel.

But the standard does not specify whether the messages at steps S1 and A4 must be transported over the same SSL/TLS channel.

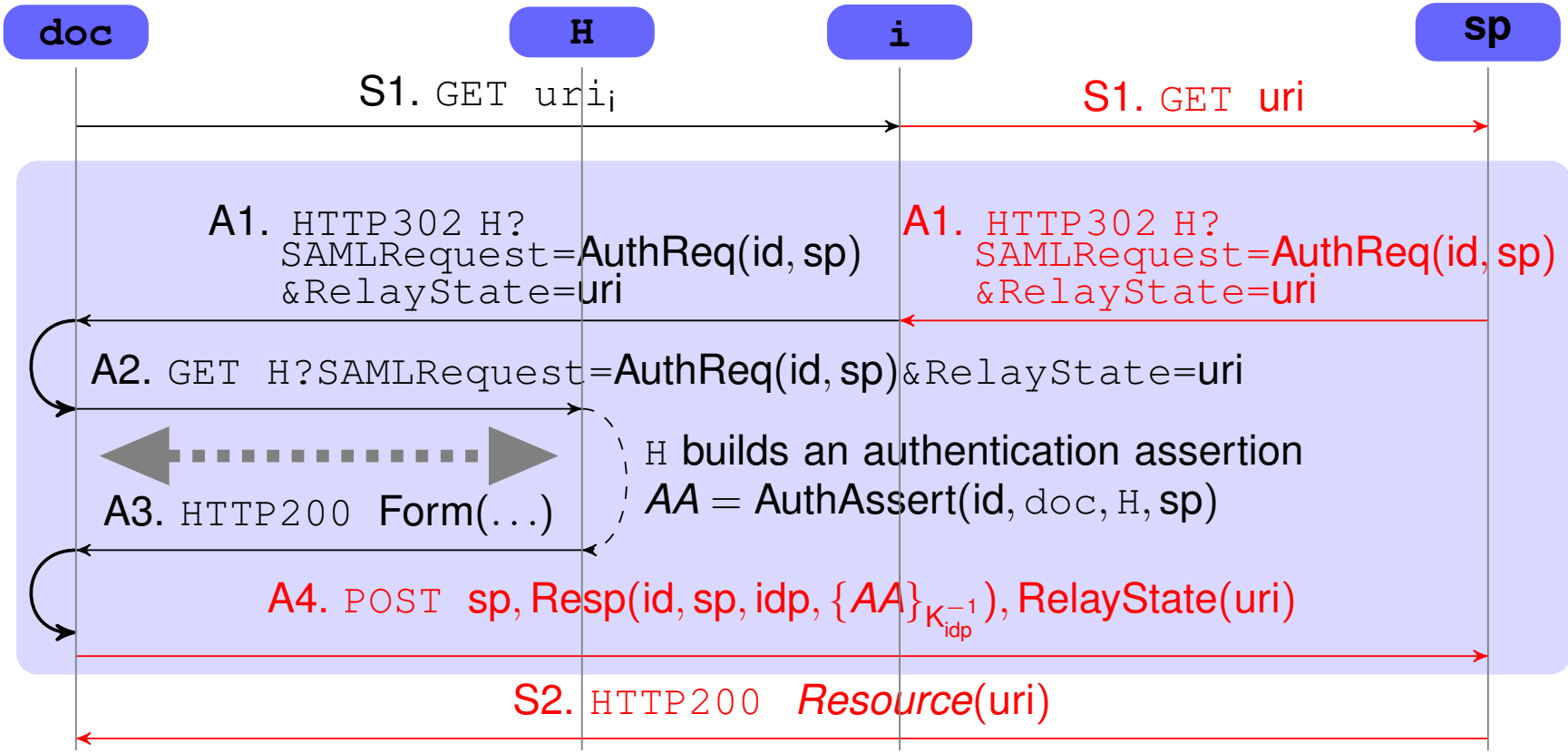
An Authentication Flaw in SAML SSO

Reuse of the SSL/TLS channel apparently the most natural option, but difficult to achieve:

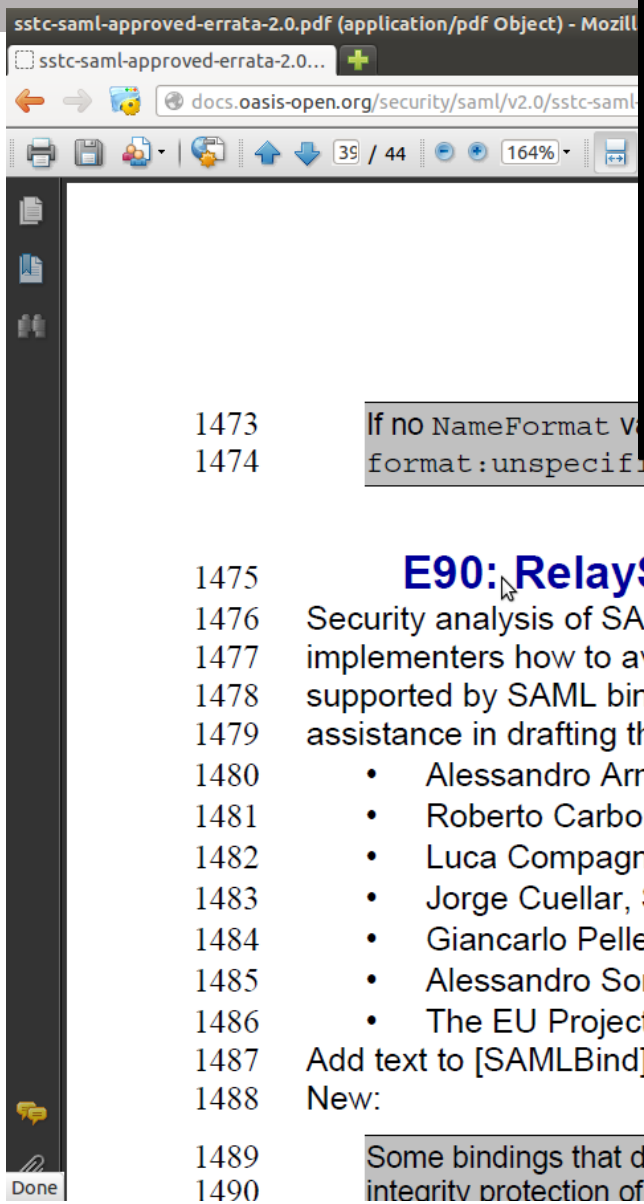
- **Resuming SSL/TLS sessions.**
  - the underlying TCP connection might be terminated,
  - an SSL server could not resume a previous session, or
  - the browser may very renegotiates the SSL session.
- **Software modularity.** The SW module that handles SAML messages may not have access to info of SSL/TLS.
- **Distributed SPs.** The SAML SP may be distributed over multiple machines, e.g., for work-balancing reasons.

An Attack on SAML SSO

When run against the revised model, SATMC found the following attack:



Browser-based Se



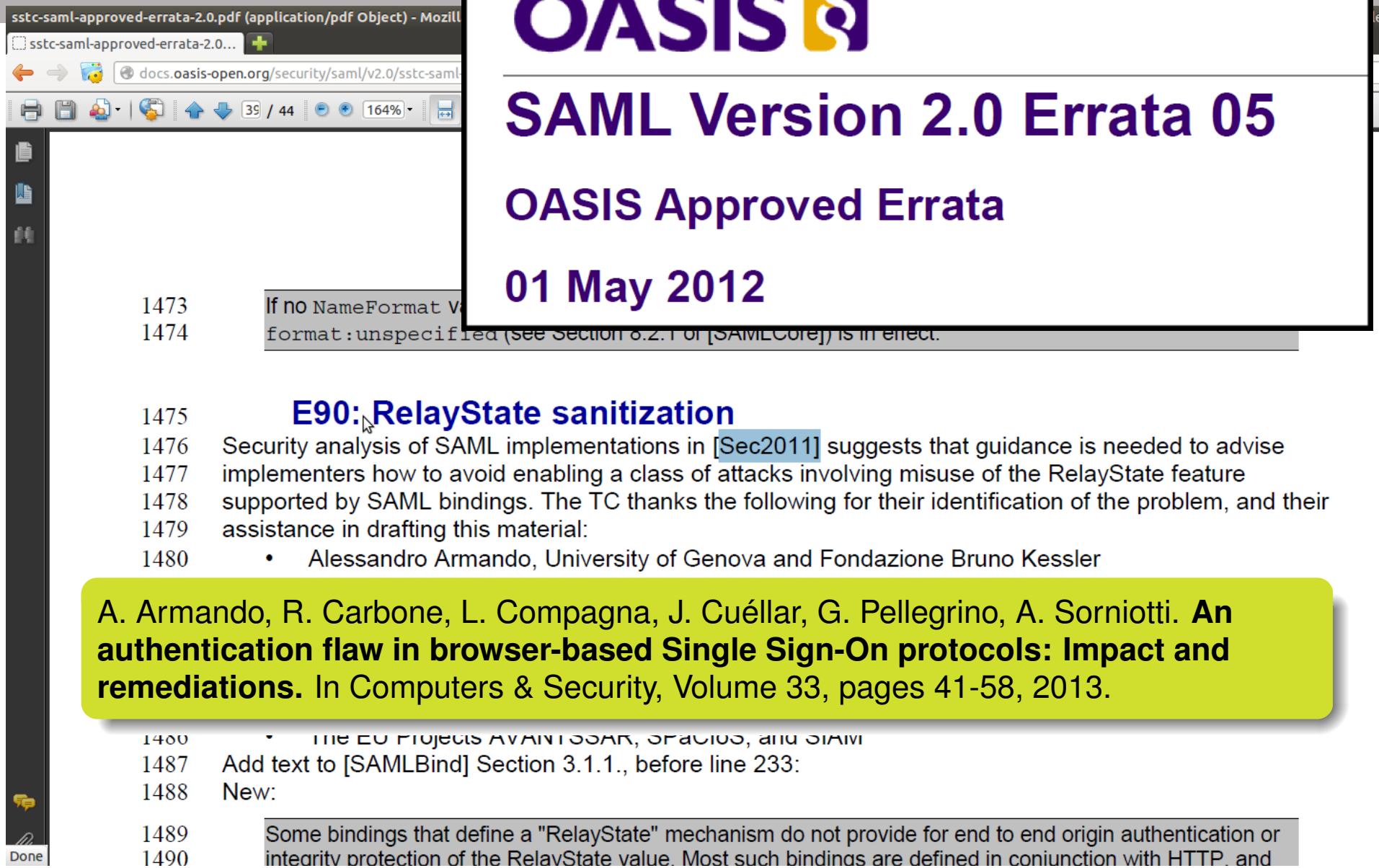
OASIS

SAML Version 2.0 Errata 05

OASIS Approved Errata

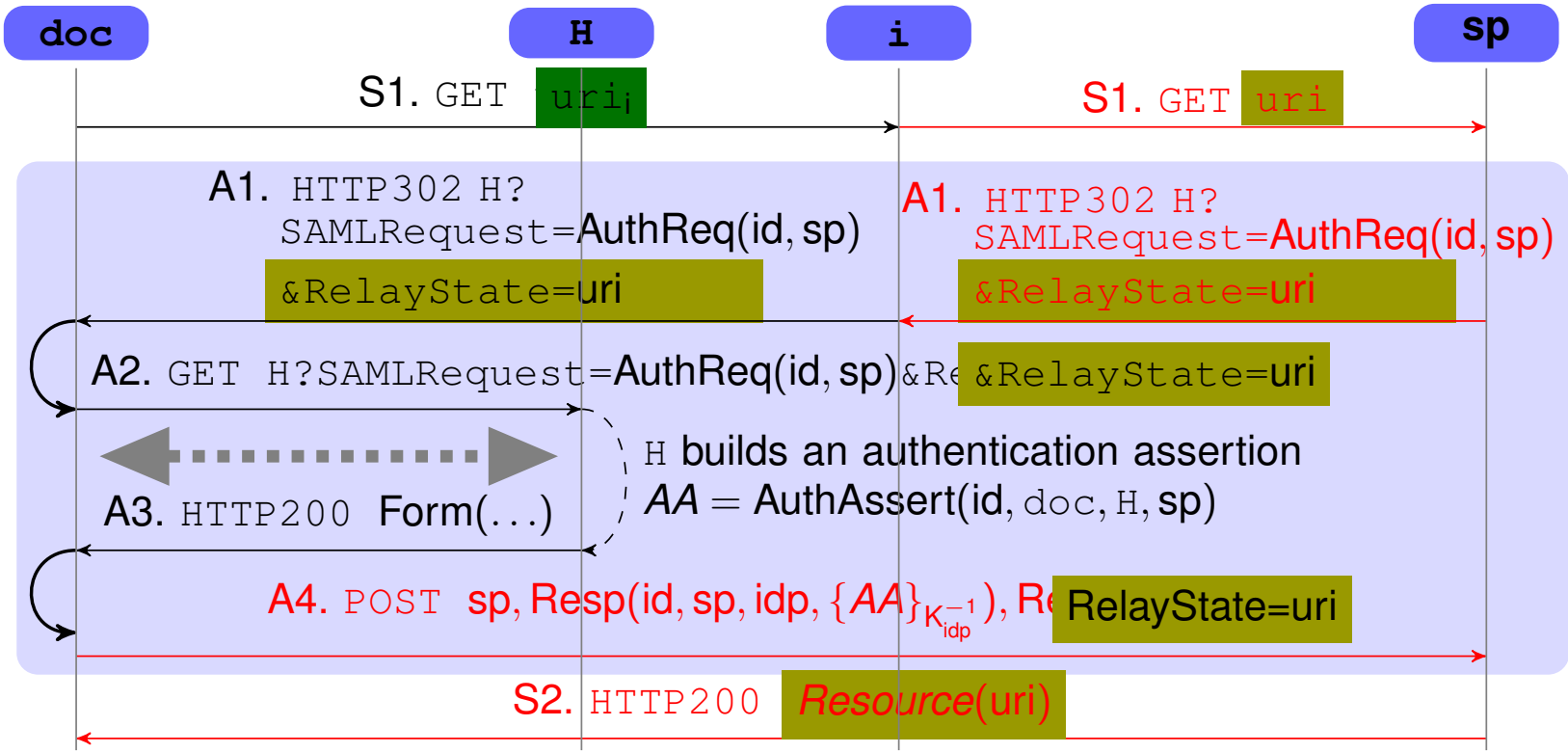
01 May 2012

Browser-based Se



A. Armando, R. Carbone, L. Compagna, J. Cuéllar, G. Pellegrino, A. Sorniotti. **An authentication flaw in browser-based Single Sign-On protocols: Impact and remediations.** In Computers & Security, Volume 33, pages 41-58, 2013.

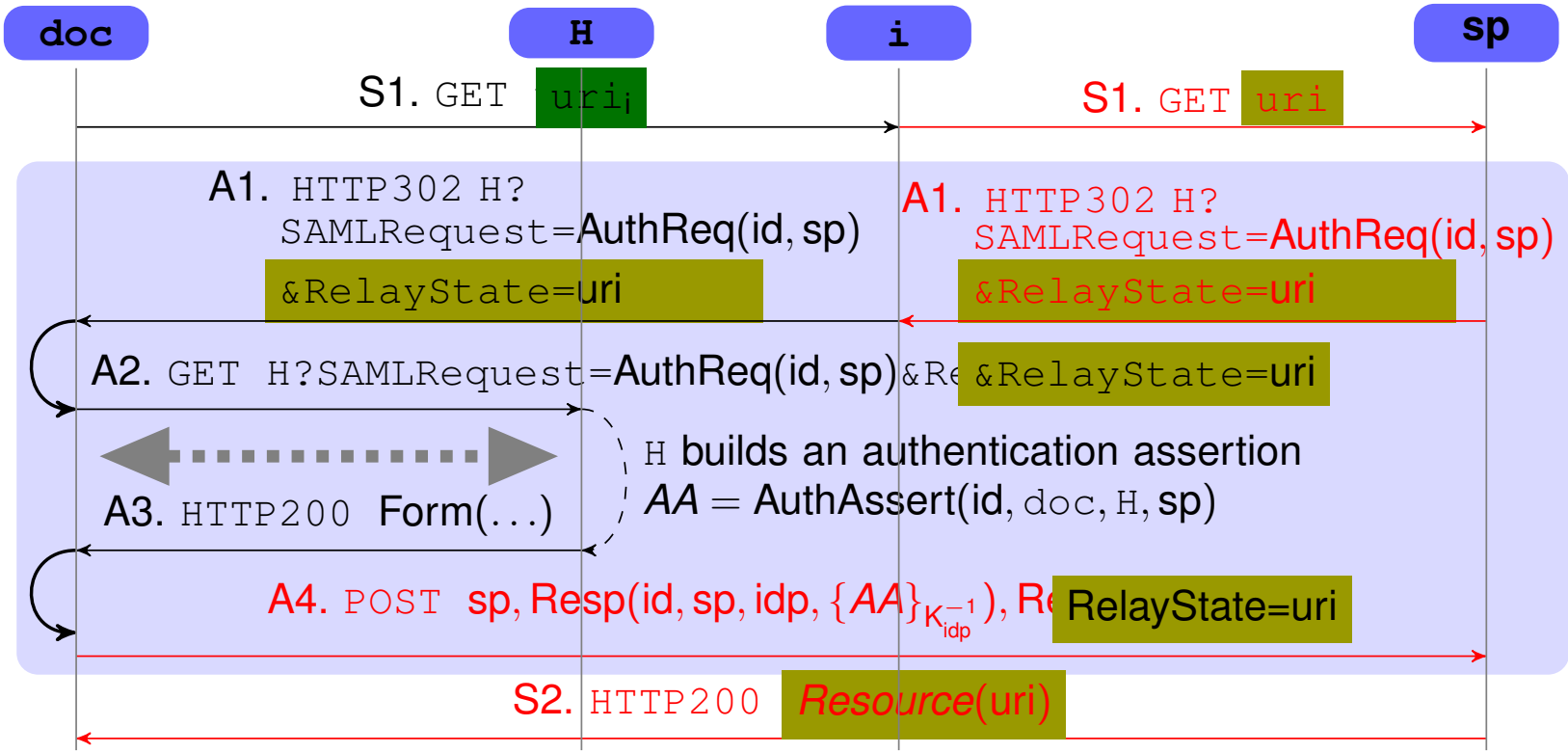
Exploiting the Vulnerability



Delivery of unrequested resource

Force C to receive a different resource from that initially requested.

Exploiting the Vulnerability

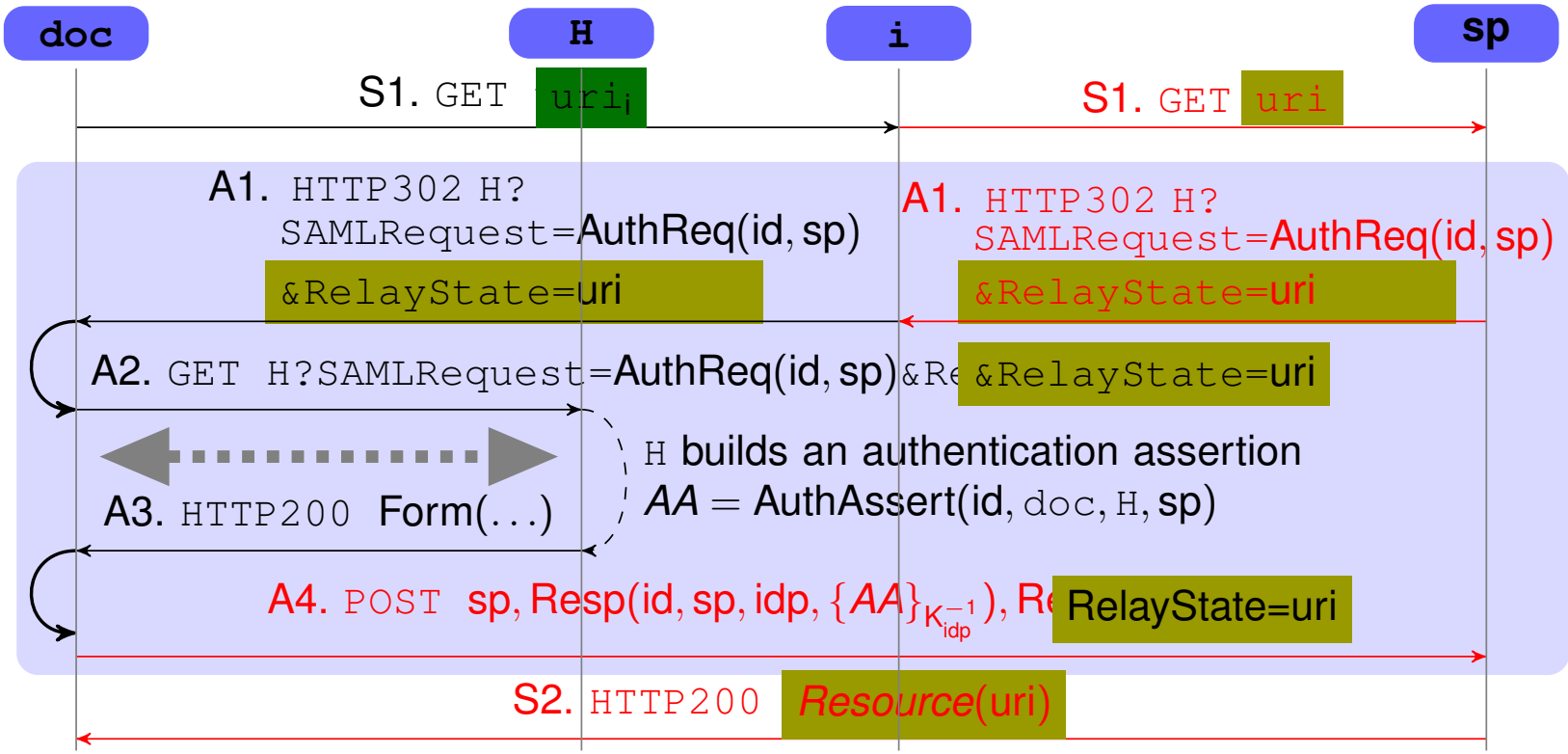


Launching pad for XSRF

URI contains a URL-encoded command (e.g. a request to change of some settings).

Even more pernicious than classic XSRF, because XSRF requires C to have an active session with SP, which is not the case here.

Exploiting the Vulnerability



Launching pad for XSS

`RelayState` exposed to injection of malicious code. Although the standard recommends to protect the integrity of this field, this often is not the case.

Impact of the vulnerability on the Google Apps

Our analysis of the SAML-based SSO for Google Apps showed that:

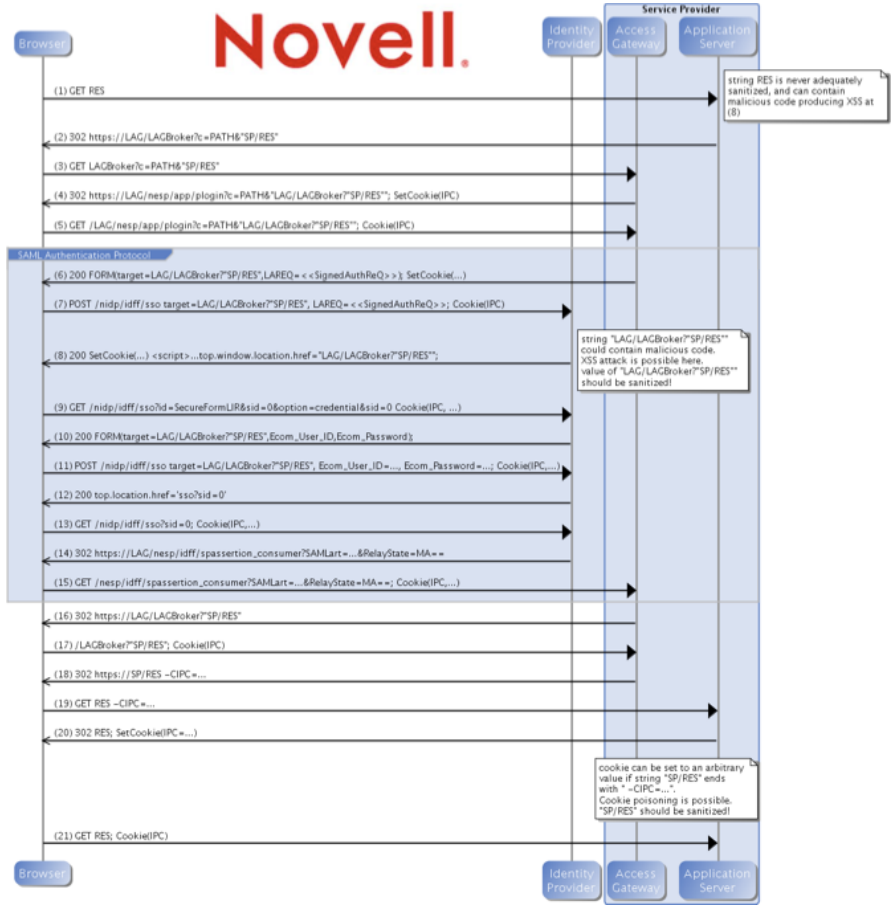
- `RelayState` was not sanitized and SAML SSO served as a launching pad for XSS.
- A malicious SP could force C to consume a resource from Google, for instance, visiting any page of the gmail service.
- A malicious SP could steal the cookies for the Google domain through XSS and could impersonate C on any Google application.

Impact of the vulnerability on simpleSAMLphp

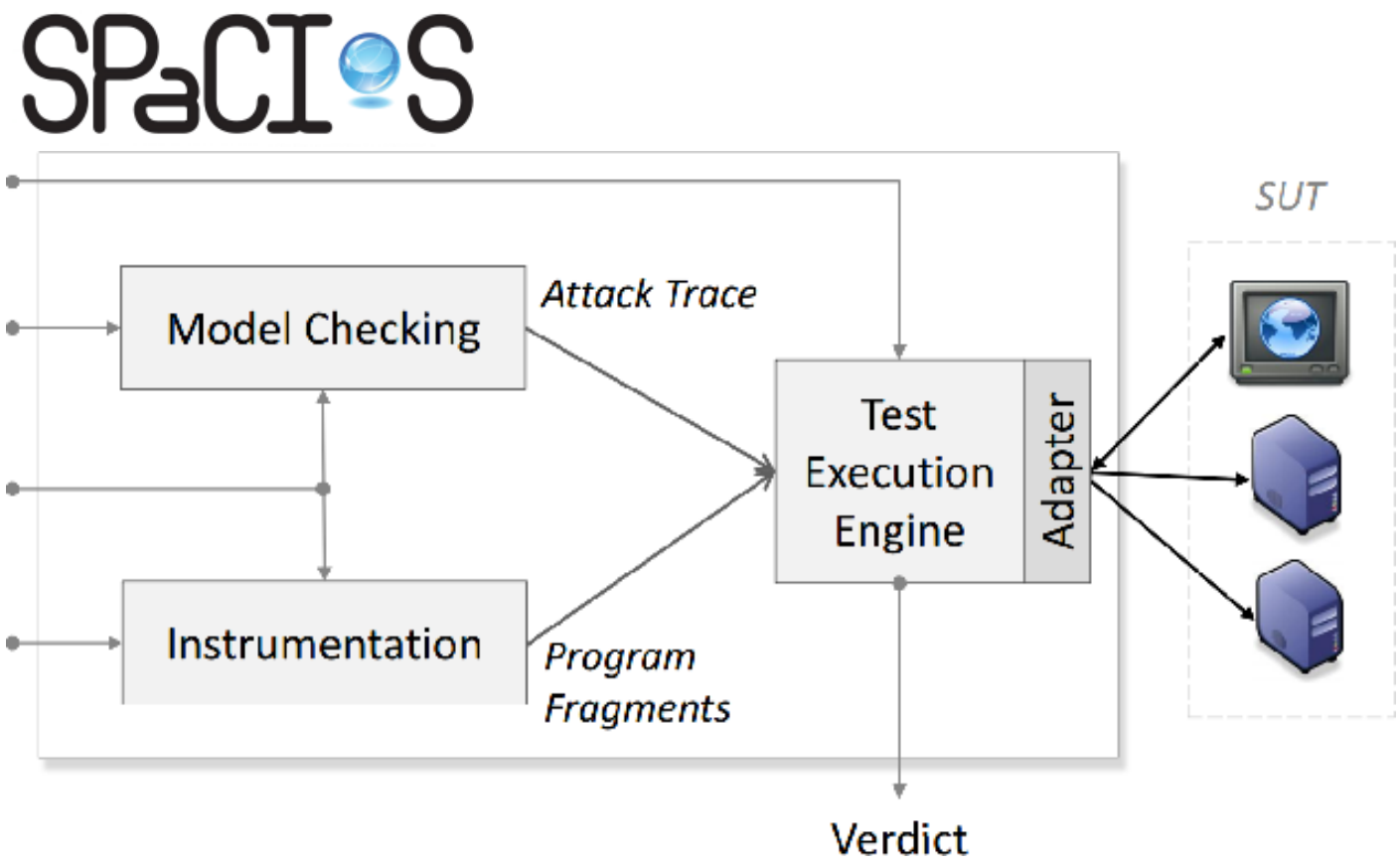
- The SimpleSAMLphp stores the initially requested URI into the URL parameter `ReturnTo`.
- Although this field is not sanitized, no XSS could be mounted.
- The SP running SimpleSAMLphp use cookies that block the exploitation.

Impact of the vulnerability on Novell Access Manager

- URI not associated with the `RelayState` field as mandated by the standard, but passed as URL-encoded parameter which was not sanitized by the SP.
- XSS attack was possible.



From Model Checking to Automated Security Testcase Generation and Execution

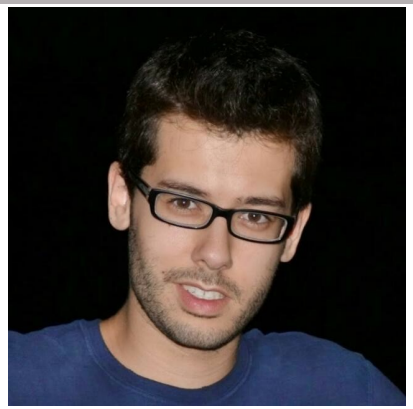


Ongoing Work...

Security Threat Identification and Testing (STIATE)

- EIT ICT Labs Activity 2014
- **Goal:** Bring the results of research projects to the market!
  - The STIATE Toolkit: an industrial strength API for model checking and automated testcase generation and execution (FBK) and front-end (SAP)
  - Methodology and guidance document for using the STIATE technology as part of Common Criteria developments. (DFKI)
  - Industry migration through application to uses cases of industrial complexity. (SAP, Reply)
  - Market solution toolkit to be market ready with consistent go to market strategy (Reply)

Another Use Case for STIATE...



- *Secure Call Authorization* (*SCA*) is a commercial solution for multi-factor and two-channel authentication developed by AliasLab S.p.A.
  - user's mobile phone (second factor)
  - GSM/3G communication infrastructure (second channel)
- **Goal:** Formal Analysis of *SCA* using the STIATE Toolkit.

Automatic Analysis of Browser-based Security Protocols

SECENTIS

A European Industrial Doctorate on Security and Trust

- Topic V, supervisors:
  R. Carbone and L. Compagna (SAP)
- The attacks described cannot be detected by the state-of-the-art penetration testing tools.
- **Goal:** Extend penetration testing tools!
- “Issue” of previous approaches: generation of the model

Question: Is it possible to detect the previous attacks without even specifying the model of the protocol?

Automatic Analysis of Browser-based Security Protocols

SECENTIS

A European Industrial Doctorate on Security and Trust

- Topic V, supervisors:
  R. Carbone and L. Compagna (SAP)
- The attacks described cannot be detected by the state-of-the-art penetration testing tools.
- **Goal:** Extend penetration testing tools!
- “Issue” of previous approaches: generation of the model

Question: Is it possible to detect the previous attacks without even specifying the model of the protocol?

Automatic Analysis of Browser-based Security Protocols

SECENTIS
A European Industrial Doctorate on Security and Trust

- Topic V, supervisors:
  R. Carbone and L. Compagna (SAP)
- The attacks described cannot be detected by the state-of-the-art penetration testing tools.
- **Goal:** Extend penetration testing tools!
- “Issue” of previous approaches: generation of the model

Question: Is it possible to detect the previous attacks without even specifying the model of the protocol?
Answer: Avinash talk !

Outline

Conclusions

- Security protocols play pivotal role e.g. in web applications (SAML SSO, OpenID, OAuth, ...)
- Formal modeling and automatic analysis of security protocols can help unveil serious flaws and get the model right
- **It works!** Vulnerabilities detected on a number of important protocols:
  ASW, SAML 2.0 SSO Profile, Google's SAML-based SSO for Google Apps, Novell Access Manager, Strong Authentication protocols, ...
- Ongoing Work: Annibale (STIATE), Federico (AliasLab), Avinash (SECENTIS)

Thank you!