# Enhancing Web Application Security Using Another Authentication Factor

Karen Lu and Asad Ali

Gemalto, Inc.

Technology & Innovations

Austin, TX, USA

# Overview

- Introduction
- Current State
- Smart Cards
- Two-Factor Authentication Techniques
- Implementation and deployment
- Conclusion

# Introduction

- Web applications are part of our daily lives
  - Work, communication; social;
  - Banking; shopping; entrainment
- Security breaches & online identity thefts are on the rise
  - Client side, server side
- Security is critical for high value transactions
  - Account login
  - Transaction authorization
  - Document signing
- User authentication is the door keeper
- Economics of security

- Introduction
- **Current State**
- Smart Cards
- Two-Factor Authentication Techniques
- Implementation and deployment
- Conclusion

# Current State

- Most websites use single factor authentication - *Password*
- Passwords are universally accepted as weak
- What does it take to break a password? *

| Numerals | 0123456789 | | | | | | |
|---|---|---|---|---|---|---|---|
| Password | | Class of Attack | | | | | |
| Length | Combinations | Class A | Class B | Class C | Class D | Class E | Class F |
| 2 | 100 | Instant | Instant | Instant | Instant | Instant | Instant |
| 3 | 1000 | Instant | Instant | Instant | Instant | Instant | Instant |
| 4 | 10,000 | Instant | Instant | Instant | Instant | Instant | Instant |
| 5 | 100,000 | 10 Secs | Instant | Instant | Instant | Instant | Instant |
| 6 | 1 Million | 1½ Mins | 10 Seconds | Instant | Instant | Instant | Instant |
| 7 | 10 Million | 17 Mins | 1½ Mins | 1½ Mins | Instant | Instant | Instant |
| 8 | 100 Million | 2¾ Hours | 17 Mins | 1½ Mins | 10 Seconds | Instant | Instant |
| 9 | 1000 Million | 28 Hours | 2¾ Hours | 17 Mins | 1½ Mins | 10 Seconds | Instant |

| Mixed Alpha and Numerals | 0123456789AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz | | | | | | |
|---|---|---|---|---|---|---|---|
| Password | | Class of Attack | | | | | |
| Length | Combinations | Class A | Class B | Class C | Class D | Class E | Class F |
| 2 | 3,844 | Instant | Instant | Instant | Instant | Instant | Instant |
| 3 | 238,328 | 23 Secs | < 3 Secs | Instant | Instant | Instant | Instant |
| 4 | 15 Million | 24½ Mins | 2½ Mins | 15 Secs | < 2 Secs | Instant | Instant |
| 5 | 916 Million | 1 Day | 2½ Hours | 15¼ Mins | 1½ Mins | 9 Secs | Instant |
| 6 | 57 Billion | 66 Days | 6½ Days | 16 Hours | 1½ Hours | 9½ Mins | 56 Secs |
| 7 | 3.5 Trillion | 11 Years | 1 Year | 41 Days | 4 Days | 10 Hours | 58 Mins |
| 8 | 218 Trillion | 692 Years | 69¼ Years | 7 Years | 253 Days | 25¼ Days | 60½ Hours |

* http://www.lockdown.co.uk

# Current State (Cont.)

- Examples
  - One major breach lead to release of 32 million passwords *
  - Nearly 50% of users used names, slang words, dictionary words or trivial password *
- OWASP Top 10 Web Application Security Risks
  - A3: Broken Authentication and Session Management
  - Threat agents: attackers, users, insiders
  - Impact: Severe
- Impact of security breaches
  - Financial loss and pain to institutions and customers
  - Loss of key assets, reputation damage, loss of compliance

* http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf

# Current State: Single Sign-On

- Usability of username/password
  - Easy to use
  - Too many passwords to remember
  - Reuse passwords or use simple passwords
- Single Sign-On
  - Remember one password instead of dozens
  - Convenient for users
  - Easier to secure one system than to do for many
- Authentication is the key
  - The system depends on the strength of the authentication
  - Most still use username and password

# Single Sign-On

- Even greater need to strength SSO authentication
  - Break one, break all


* http://themarketingguy.files.wordpress.com

# Multi-Factor Authentication

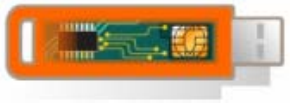
- What you know
  - password, passphrase, mother's maiden name
- What you have
  - Smart card, OTP token
- What you are or what you do - biometrics
  - Iris, finger print, face, voice, typing dynamics
- Authentications using more than one factors are called strong authentications
- How to add “what you have” factor to provide strong authentication to web applications?

- Introduction
- Current State
- **Smart Cards**
- Two-Factor Authentication Techniques
- Implementation and deployment
- Conclusion

# Smart Cards

- What is a smart card?
  - A small plastic card with an embedded microprocessor
  - secure memories; ROM, Flash, RAM
  - Hardware cryptographic engine
- Secure, portable, and tamper-resistant computer
- Multiple form factors…



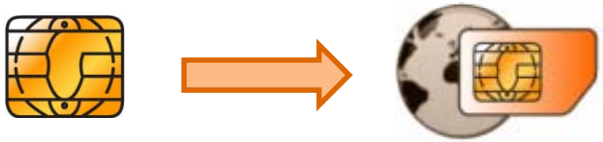
# Smart Cards (Cont.)

- Applications
  - Access control (physical, logical – e.g. Windows logon)
  - Identity (citizen cards, passports, ID cards)
  - Subscriber identification modules (SIM)
  - Banking
  - Etc.



- Smart-card-based USB tokens
  - Embedded smart card
  - Flash memory

- Using smart cards for web applications is a natural extension



# Challenges of Using Smart Cards in Web Applications

- Communications
  - Smart card communication standard
    - PC/SC – supported by all major operating systems
  - Middleware
  - Web browser connection
- Usability
  - User interface not coupled to web application
  - Web application does not have control over the user interaction
  - Terminology not understood by non-technical folks
- Different architectures for browsers / platforms
  - Custom middleware implementations
  - Not available for all platforms
  - Software upgrade issues
  - End user installation issues

- Introduction
- Current State
- Smart Cards
- **Two-Factor Authentication Techniques**
- Implementation and deployment
- Conclusion

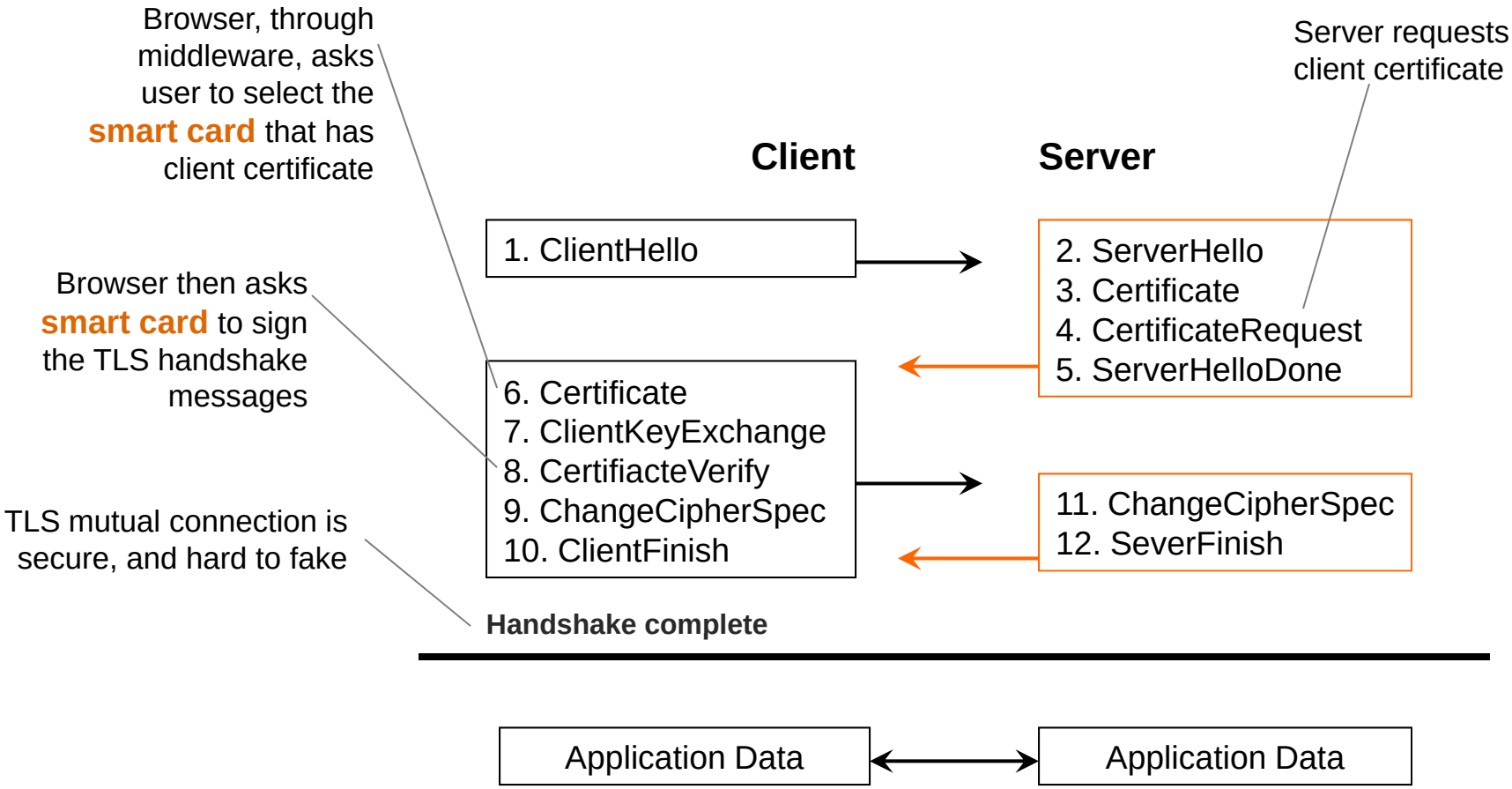
# Smart Cards for Web Applications

- Two-Factor Strong Authentications
  - What you know: PIN to the smart card
  - What you have: smart card or smart-card-based token
- Authentication methods
  - TLS mutual authentication
  - One Time Password authentication
  - PKI Certificate-based authentication
- Single Sign-On
  - SAML
  - OpenID

# TLS Mutual Authentication

- Smart card holds a X.509 certificate and the corresponding {public key, private key}
- The user registers the security token (smart card) with the web browser
- HTTPS connection from a web browser to a web server with TLS mutual authentication
- Require Middleware
  - CAPI – Cryptographic API – Windows OS
  - CDSA – Common Data Security Architecture – Mac OS X
  - PKCS#11 – API for cryptographic tokens – Firefox, OpenSSL

# TLS Mutual Authentication (Cont.)



# One-Time Password Authentication

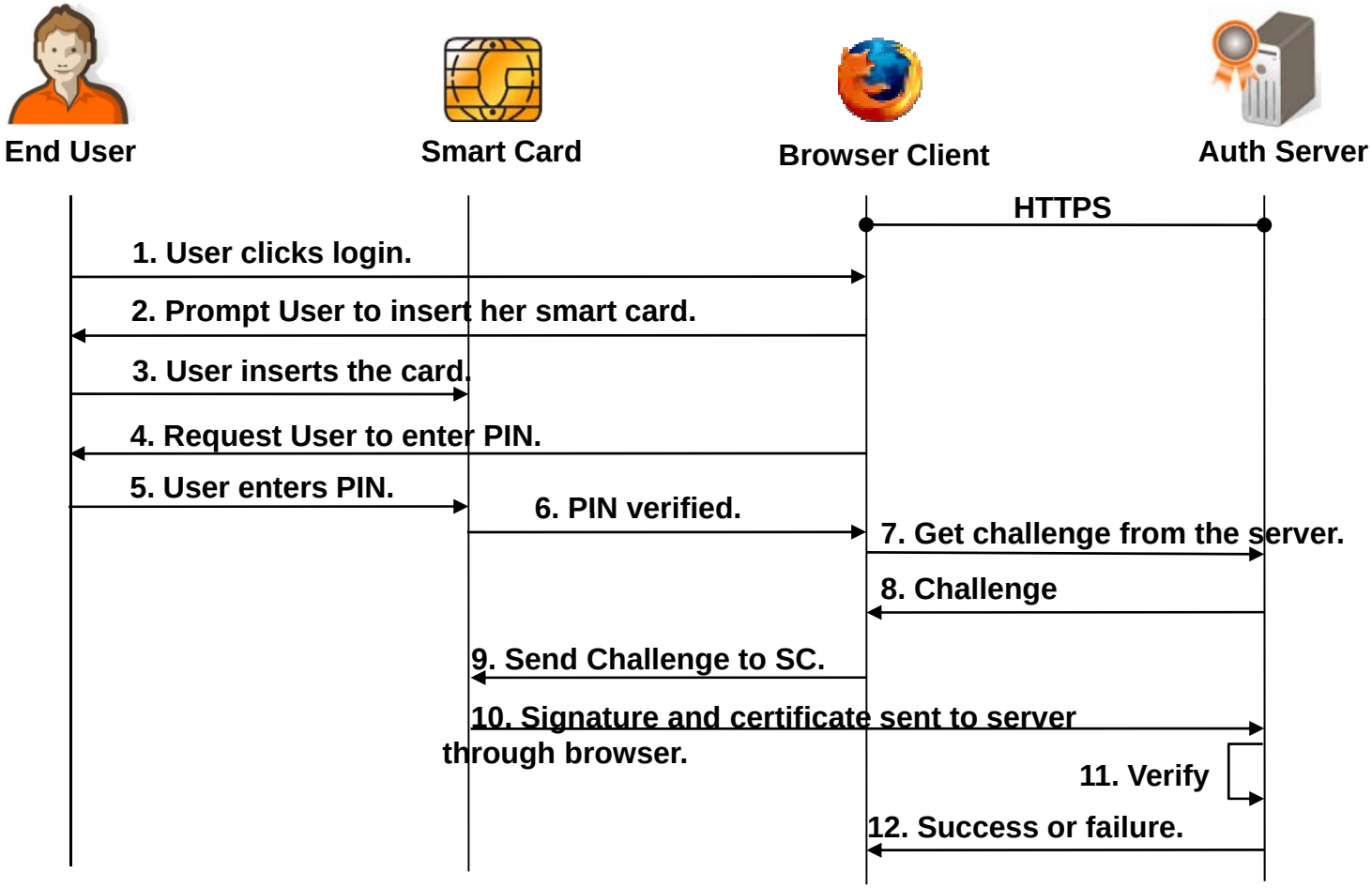
- What is One-time password (OTP)?
  - As the name suggests, it is a password that is used only once
  - Used in addition to username (and password) for authentication
  - Time-based or Event-baesd: The OTP device and the OTP server synchronize through time or event based algorithm
- Usage…
  - User pushes a button on the device
  - The device generate an OTP
  - User enters the OTP value to the web page

- Security?
  - Less secure than TLS mutual authentication
  - Security improved if user forced to enter a PIN to use the OTP token
- Usability?
  - Simple to use; no setup on the client side

# PKI Certificate-based Authentication

- Smart card holds a X.509 certificate and the corresponding {public key, private key}
- The server sends a random challenge
- The smart card digitally signs the challenge using the private key stored inside the card
- The smart card sends the signature (response) and its certificate to the server

# PKI Authentication Example (Cont.)



# PKI Authentication - *SConnect*

- SConnect is a web-browser-based approach
  - Web browser extensions
    - IE, Firefox, Safari, Chrome, Opera
    - Windows, Linux, and Mac OS X
  - Javascript API
  - Based on PC/SC – no middleware
- Build-in security features
  - Force HTTPS
  - Server verification
  - Connection key linking to server's SSL certificate
  - User consent for smart card access
- For the first time use, the user is prompted to download a web browser plug-in

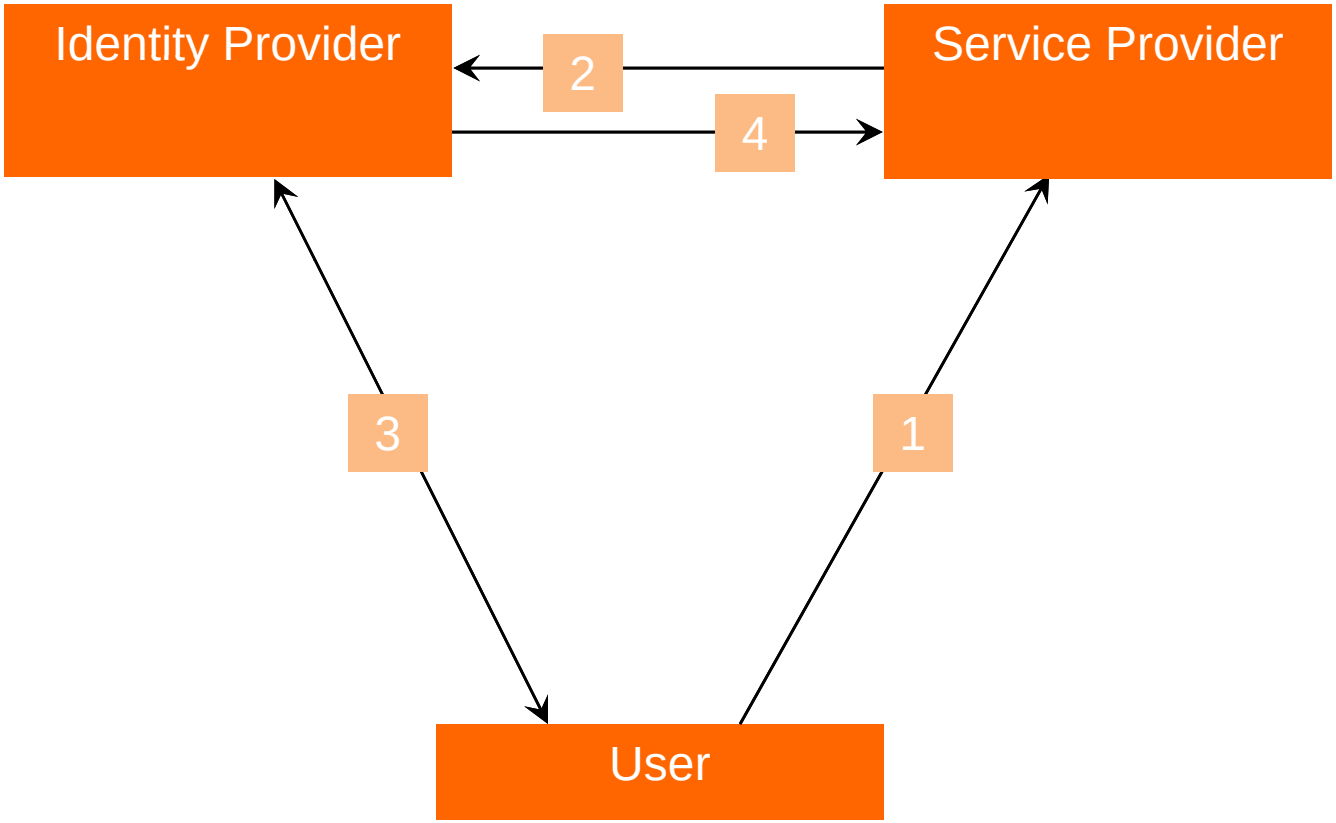
# Single Sign On / Federation

- Two aspects
  - Login once and access multiple service providers
  - Use one login credential to login to multiple service provider
- Benefit
  - Convenient to users – use as few credentials as possible
  - Service providers – delegate authentication to identity provider
- Authentication is the key
- (*de facto*) Standards
  - SAML – OASIS standard
  - OpenID – Open standard

# Actors in Single-Sign On

- User
  - Entity that can acquire a federated identity
  - Capable of making decisions
  - User, Group of individuals, a corporation, a legal entity etc
- Identity Provider
  - Creates, manages and maintains the identity information of Principal
  - Provides user authentication to other service providers within a circle of trust
- Service Provider
  - An entity that provides services to the user

# Single Sign-On (Cont.)



# Standard Bodies

- Kantara Initiative (formerly Liberty Alliance)

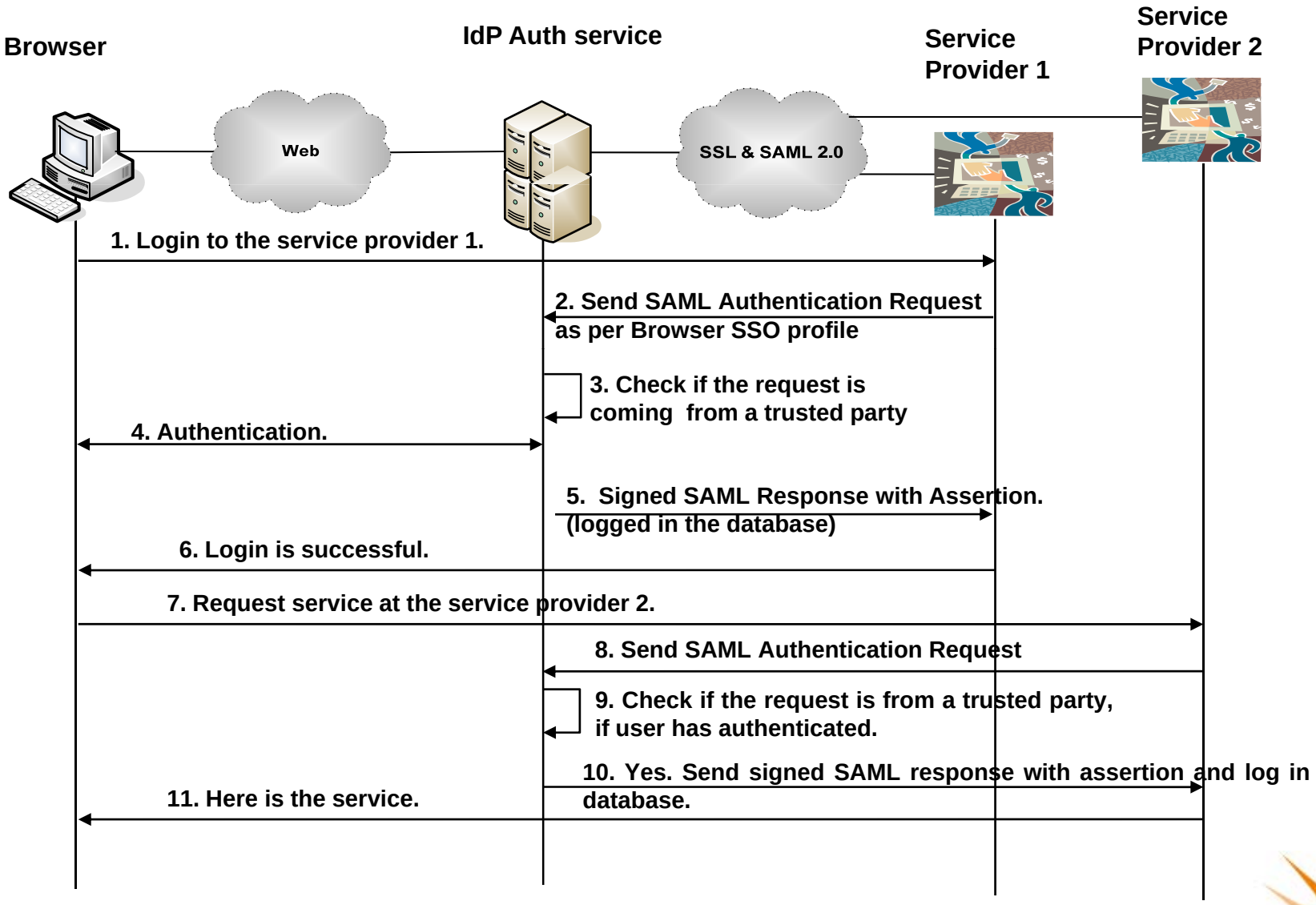

- OASIS (Organization for the Advancement of Structured Information Standards)
  - SAML
  - WS-*


- OpenID

# Security Assertion Markup Language

- Current Version is SAML 2.0 – OASIS Standard
- Consolidate earlier work done in Liberty Alliance
- Assertions, Protocols, and Bindings
  - Generated assertion would contain user's x509 certificate and configured attributes
- Getting adoption in the Government space
- Flexible and Extensible framework
- SAML 2.0: Browser Single Sign-On profile
  - Redirect Binding

# SAML SSO

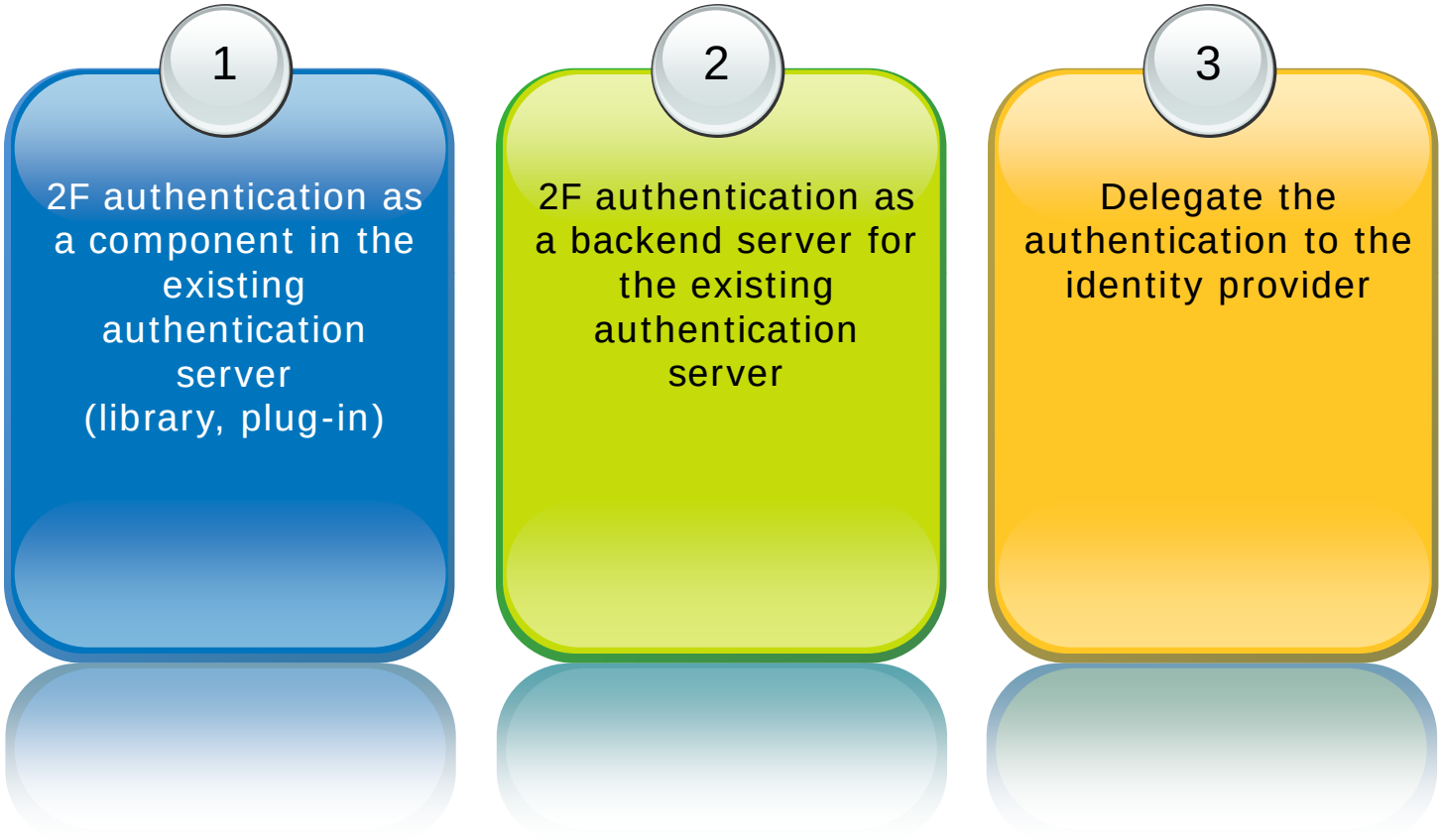

Browser
IdP Auth service
Service Provider 1
Service Provider 2
Web
SSL & SAML 2.0
1. Login to the service provider 1.
2. Send SAML Authentication Request as per Browser SSO profile
3. Check if the request is coming from a trusted party
4. Authentication.
5. Signed SAML Response with Assertion. (logged in the database)
6. Login is successful.
7. Request service at the service provider 2.
8. Send SAML Authentication Request
9. Check if the request is from a trusted party, if user has authenticated.
10. Yes. Send signed SAML response with assertion and log in database.
11. Here is the service.

# OpenID

- Developed by OpenID Foundation
- Open, decentralized framework for user-centric digital identity management.
- Current version: OpenID Authentication 2.0
- Actors
  - OpenID Provider
  - OpenID User
  - OpenID Service Provider
- User chooses which OpenID provider to use when login to a service provider that supports OpenID.
- Mostly used for low-value transaction websites.
- US Government Services Administration's pilot adoption of OpenID for Open Government

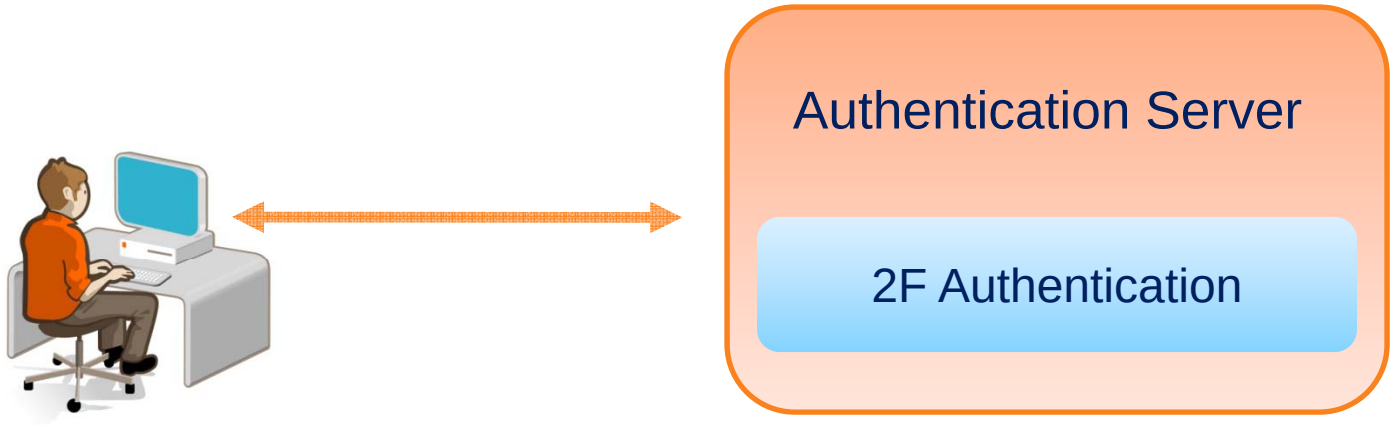
- Introduction
- Current State
- Smart Cards
- Two-Factor Authentication Techniques
- **Implementation and deployment**
- Conclusion

# Implementation and Deployment



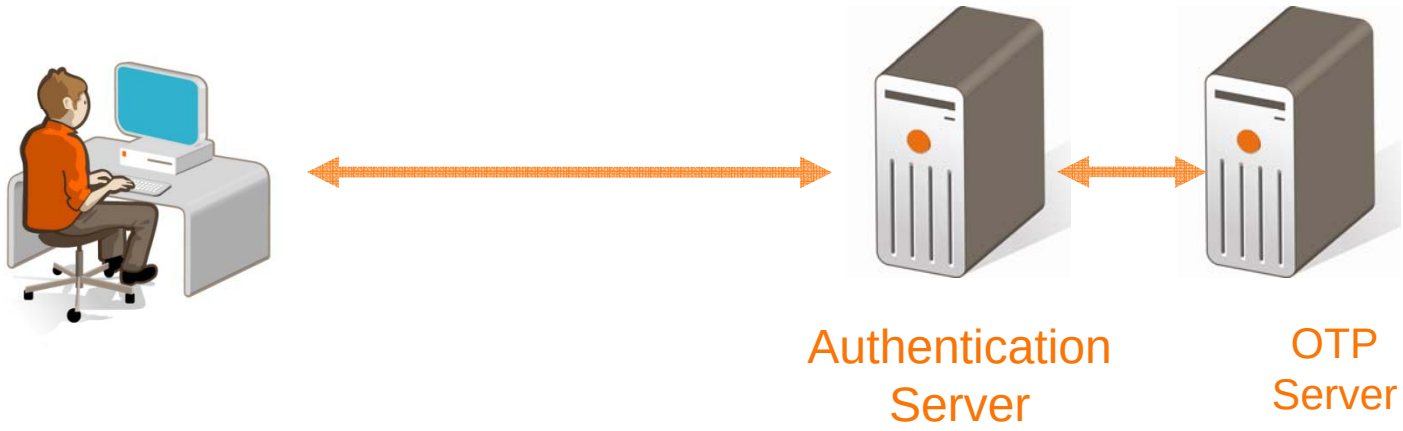
# Component

- The two-factor authentication can be a component of a website's existing authentication server.



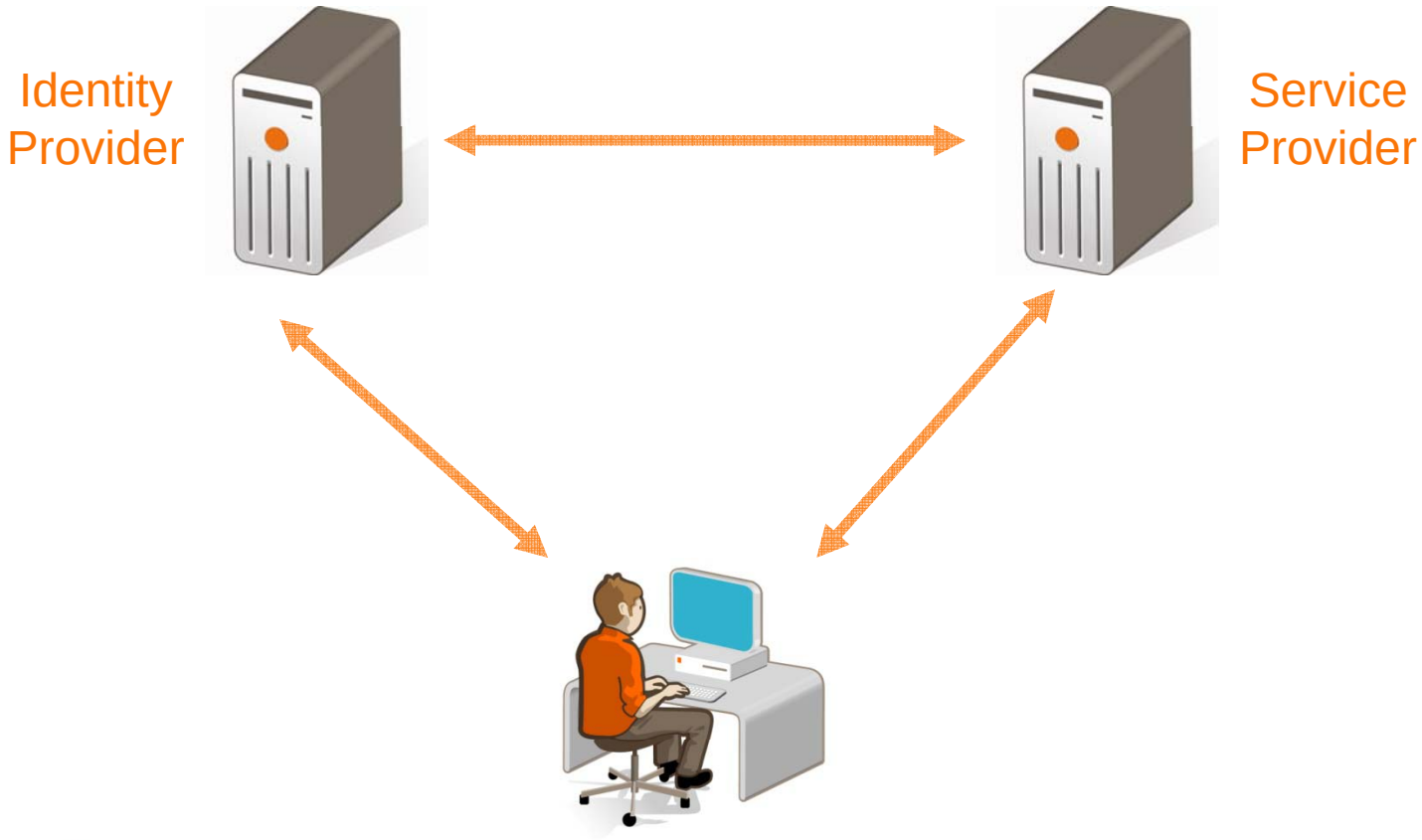
# Backend Server

- The two-factor authentication can be a backend server of a website's existing authentication server.
- Example: Existing server handles username/password, OTP server handles OTP.



# Delegate to Identity Provider

- Service providers redirect the user to the identity provider for authentication (SAML, OpenID, Facebook Connect)



# Conclusions

- Internet Security…
  - Single-factor, knowledge based authentication is weak
  - Really “eliminate password”, not just push it downstream
  - Adoption occurs only when increased security makes economic sense
- Smart Cards…
  - Traditional roles are necessary, but not always sufficient for future expansion into the increasingly digital world
  - Provide enhanced security while working within the constraints of established enterprise and government frameworks
- Consumer space…
  - More challenging: a wider spectrum of operating environments
  - Users do not always like to carry additional tokens
  - Need to make devices multi-functional, or leverage existing device deployment
- The real challenge…
  - Love is simple, trust is hard