Remote Authority Services through CDSA

Denise Ecklund

The Open Group Meeting

April 29, 1999

intel. psd Platform Security Division

Agenda

- CA Services Before Spec Revision
- Evolving Requirements
- Strategy to Address Requirements
- Design Overview
- Questions and Feedback

Original CDSA Support for CA Services

- One service was provided, “Issue a Certificate”
- Defined Registration Authority functions
  - RegistrationFormRequest(…)
  - RegistrationFormSubmit(…)
- Supported asynchronous CA service
  - CertRequest(…)
  - CertRetrieve(…)
- Input parameters supported some options
  - Selecting the CA
  - Oversigning/notarizing
  - Passing authentication credentials to the CA

Evolving Requirements

- CA offers more services
  - Revocation
  - On-line verification
  - Suspension and Release
  - Notarization
  - Reclamation (recover use of private key)
- Competing protocols shape service definitions
- External protocol and data format standards evolve without us.

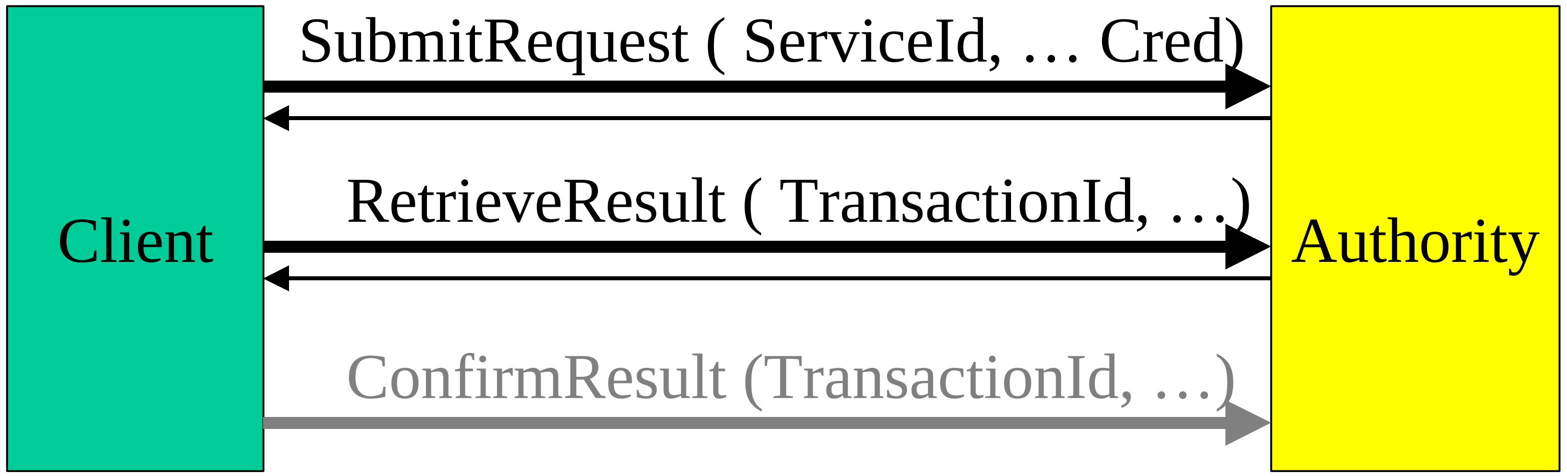
How to Keep Up?

Addressing Requirements

- Keep APIs open to
  - New CA services: CertUpgrade, …, PKCS#12
  - New protocols: CRMF over CMS?
- Design Approach
  - Retain three-party model: EE, RA, CA
  - Define a few generic, slightly extensible APIs to encapsulate existing protocols
  - Define service-specific data structures
- Anticipated Results
  - Do not perturb existing data structs and APIs
  - May need to add a new, *independent* data struct

Design Overview

Authenticated Client-to-Authority Calls


Client
SubmitRequest (ServiceId, … Cred)
RetrieveResult (TransactionId, …)
ConfirmResult (TransactionId, …)
Authority

Currently Defined ServiceIds:

- CertIssue
- Change state: { revoke, hold, release }
- Verify
- Notarize
- Reclaim
- CrlIssue

Service-specific Types and Structures

- Each CA service has
  - A unique ID
  - A set of service-specific data structures
- Service-specific structures
  - Input to SubmitRequest( )
  - Output from RetrieveResult( )
- Service-specific types
  - List of constant values such as “status” or return values

Example - CertIssue

- CSSM_TP_CertIssue_Input

  { CSPSubserviceUid;
  SubjectCertFields;
  NumberOfFields;
  MoreServiceRequests;
  ServiceControls;
  NumberOfServiceControls;
  UserCredentials }

- CSSM_TP_CertIssue_Status

  #define … UNKNOWN
  #define … OK
  #define … OKWITHCERTMODS
  #define … OKWITHSERVICEMODS
  #define … REJECTED
  #define … NOT_AUTHORIZED
  #define … WILL_BE_REVOKED

- CSSM_TP_CertIssue_Output

  { IssueStatus;
  CertGroup;
  PerformedServiceRequests}

intel. psd Platform Security Division

Example - CertChange

- CSSM_TP_CertChange_Action

  #define … NONE
  #define … REVOKE
  #define … HOLD
  #define … RELEASE

- CSSM_TP_CertChange_Input

  { Action;
  Reason;
  Cert;
  ChangeInfo;
  StartTime;
  CallerCredentials }

- CSSM_TP_CertChange_Status

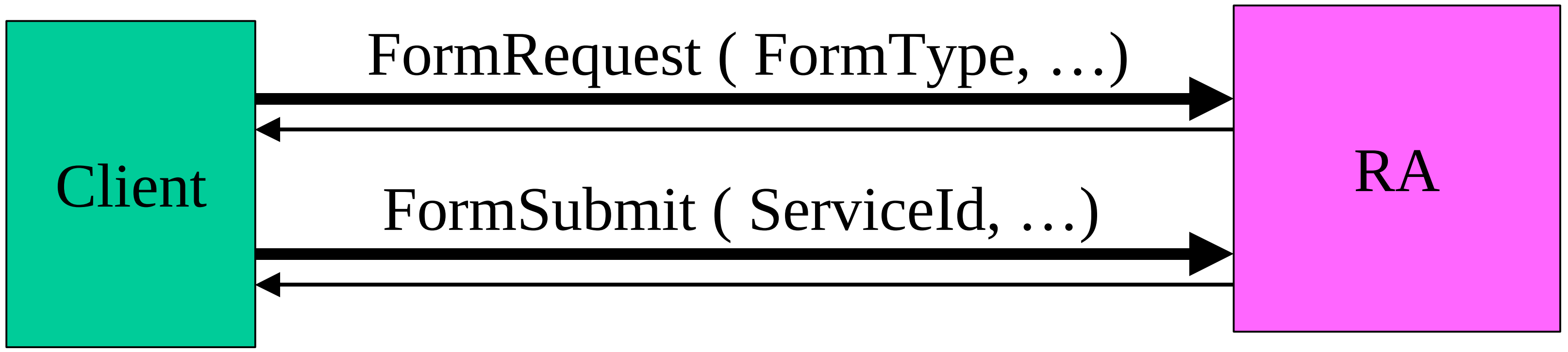
  #define … UNKNOWN
  #define … OK
  #define … OKWITHNEWTIME
  #define … WRONGCA
  #define … REJECTED
  #define … NOT_AUTHORIZED

- CSSM_TP_CertChange_Output

  { ChangeStatus;
  ChangeInfo }

intel psd Platform Security Division

“Free” Client-to-RA Calls



Currently Defined FormTypes:

Generic
Registration

Summary

- Technical aspects of the approach
  - Encapsulates a small range of different protocols
  - Same old approach to asynchrony
  - Provides "good" insulation from other changing standards embraced and used within CDSA
  - Differs from the old approach; now service type is identified by parameter rather than function name

End Result:

A consistent set of remote service interfaces
that can encapsulate other industry standard protocols
and data formats for accessing CA services.