Independent Object Protection GSS API

Carlisle M. Adams
Paul Van Oorschot
Bell-Northern Research

Dec. 5-9, 1994
31st Internet Engineering Task Force
San Jose, CA

Outline

- **Background**
  - **Generic Security Service (GSS) [RFC 1508]**
- **IOP Motivation and Goals**
- **Function Calls**
- **Implementation News**
- **Conclusions**

Background

- **Generic Security Service (GSS)**
  - **an interface specification**
  - **defines a set of calls for services which applications developers can use to incorporate security**
  - **although "generic", this API is targeted toward session-based (on-line) application peers**
  - **a security context is established between a pair of communicating applications**
    - » **both parties are required in order to estab. context**
    - » **message protection/unprotection has no meaning outside context**

Outline

- **Background**
  - **Generic Security Service (GSS) [RFC 1508]**
- ⇨ **IOP Motivation and Goals**
- **Function Calls**
- **Implementation News**
- **Conclusions**

IOP-GSS

- **Motivation:**
  - **many important applications are not session-based (e.g., e-mail)**
  - **need to protect a data object independently of any other data object**
  - **need to protect a data object independently of any on-line communication with target(s)**
- **Goals:**
  - **an extension to the current GSS**
  - **want to re-use as much of GSS as possible**
  - **want to imitate the functionality of sign and seal (getMIC and wrap), but for "off-line" environments**

Outline

- **Background**
  - **Generic Security Service (GSS) [RFC 1508]**
- **IOP Motivation and Goals**
- ⇨ **Function Calls**
- **Implementation News**
- **Conclusions**

Function Calls

- **Context management calls:**
  - IOP_Init_context()
  - IOP_Accept_context()
  - IOP_Delete_context()

  - security context is not shared on-line with peer
    - » rather, context marks beginning to end of local object processing operations
    - » no explicit concept of expiration time of context (but credentials can still expire)

Function Calls (cont'd)

- **"Per-object" calls:**
  - IOP_Protect()
  - IOP_Unprotect()

  - each call split into three parts
    - » Start_Operation() -- called to identify algs, QOP
    - » Operation() -- called once per buffer of input data
    - » End_Operation() -- called to return results

  - possible result tokens
    - » prot_token -- contains info. req'd. for target to unprot.
    - » receipt_token -- contains proof of receipt of prot. object

Function Calls (cont'd)

- **Support call:**
  - IOP_Extract_Prot_Token()

  - used to extract the prot_token from a protected object

  - token is then input to IOP_Unprotect() call

- **Some new Major Status values defined**
  - IOP_S_ ...

Outline

- **Background**
  - Generic Security Service (GSS) [RFC 1508]
- **IOP Motivation and Goals**
- **Function Calls**
- ⇨ **Implementation News**
- **Conclusions**

Implementation News

- **BNR is implementing a PEM-based IOP-GSS-API mechanism**
- **Implementation results to be discussed at April IETF**
- **Our implementation is not public domain**
  - some proprietary features / algorithms
  - high-performance versions of public algorithms

Outline

- **Background**
  - Generic Security Service (GSS) [RFC 1508]
- **IOP Motivation and Goals**
- **Function Calls**
- **Implementation News**
- ⇨ **Conclusions**

Conclusions

- IOP-GSS-API is a useful GSS API extension for "off-line" environments:
  - incorporates many of the concepts and calls of GSS
  - new context management, per-object, support calls
  - can be supported with public-key-based or secret-key-based mechanisms
- IOP-GSS-API is being implemented by BNR; we are encouraging any interested parties to do a public-domain implementation
- We offer this as a base document for further work on a store-and-forward GSS-API spec.