PEM-Based IDUP Mechanism (PIM)

Dragan Grebovich
Carlisle Adams
Bell-Northern Research
Ottawa, Canada

July 17-21, 1995
33rd Internet Engineering Task Force
Stockholm, Sweden

Introduction

- PIM is a simple specification for an IDUP mechanism which is based on PEM
- "Marriage" of the IDUP-GSS-API Internet Draft and the PEM spec. (RFC-1421)
- Internet Draft: draft-ietf-cat-pim-00.txt

Concepts

- PIM is *based on* PEM; it does not create full PEM messages
  - PIM creates full PEM header (as prot_token)
  - PIM optionally encrypts data (IDU)
  - calling application is responsible for
    - » canonicalization of IDU prior to IDUP calls
    - » PEM-encoding of (enc.) IDU subsequent to IDUP calls
    - » concatenation of PEM delimiters, CRLFs, prot_token, and "PEM-ified IDU"
- IDUP (with PIM) does *security* part of PEM; application does non-security operations
  - would be easy to build an API on top of IDUP/PIM to do full PEM messaging

Concepts (cont'd)

- Receipt generation and processing are not supported (not used in PEM)
- Evidence generation and processing are not supported (not used in PEM)
- Therefore, receipt_check, receipt_token, and evidence_token are not currently defined in PIM

Concepts (cont'd)

- Originator_info (returned by IDUP_End_Unprotect())

```
<originator-info>  ::=
  [<entity-id>] CRLF [<issuing-auth>] CRLF [<version-expiration>] CRLF
<entity-id>           ::=  1*(<ia-char>  /  ",")
<issuing-auth>        ::=  1*(<ia-char>  /  ",")
<version-expiration>  ::=  1*(<ia-char>  /  ",")
```

- Values for originator_info
  - may be read directly from prot_token (if <origid-symm> or <origid-asymm>) are present, or
  - may be extracted from the originator's certificate (if <cert> is present)

Examples

- Integrity only:

```
prot_token  =  {
  Proc-Type:  4,MIC-CLEAR
  Content-Domain:  RFC822
  Originator-Certificate:  ...
  Issuer-Certificate:  ...
  MIC-Info:  RSA-MD5,RSA,...
}

idupart  =  {
  This is the plaintext message part...
}
```

Examples (cont'd)

- Confidentiality only:

```
prot_token  =  {
  Proc-Type:  4,ENCRYPTED
  Content-Domain:  RFC822
  DEK-Info:  DES-CBC,...
  Originator-ID-Symmetric:  ...
  Recipient-ID-Symmetric:  ...
  Key-Info:  DES-ECB,RSA-MD2,...,...
}
idupart  =  {
  This is the encrypted (binary) message part...
}
```


Conclusions

- PIM is a simple, PEM-based mechanism for IDUP
  - security operations of PEM are done by PIM
  - application is responsible for all other PEM-related operations
- PIM is being implemented by BNR
  - proprietary implementation (not public domain)
  - PIM specification was done primarily to show usability of IDUP; other mechanisms (e.g., MOSS, MSP) should also be done