IDUP and SPKM:

Developing Public-Key-Based APIs and Mechanisms for Communications Security Services

Carlisle M. Adams

Bell-Northern Research

ISOC SNDSS, Feb. 22-23, 1996

Outline

- **Introduction and Terminology**
- **IETF CAT Group Activities**
- **Session: GSS / SPKM**
- **Store&Forward: IDUP / PIM**
- **Comparisons with Other Work**
- **Conclusions**

Introduction and Terminology

- **An increasing number of applications are becoming aware of a need for security**
- **Many application developers do not have the time or expertise to embed security services themselves**
- **Application Program Interfaces (APIs) offering security services can be a practical solution, given**
  - **verified security practices**
  - **ease of use**
  - **standardization**

Introduction and Terminology (cont'd)

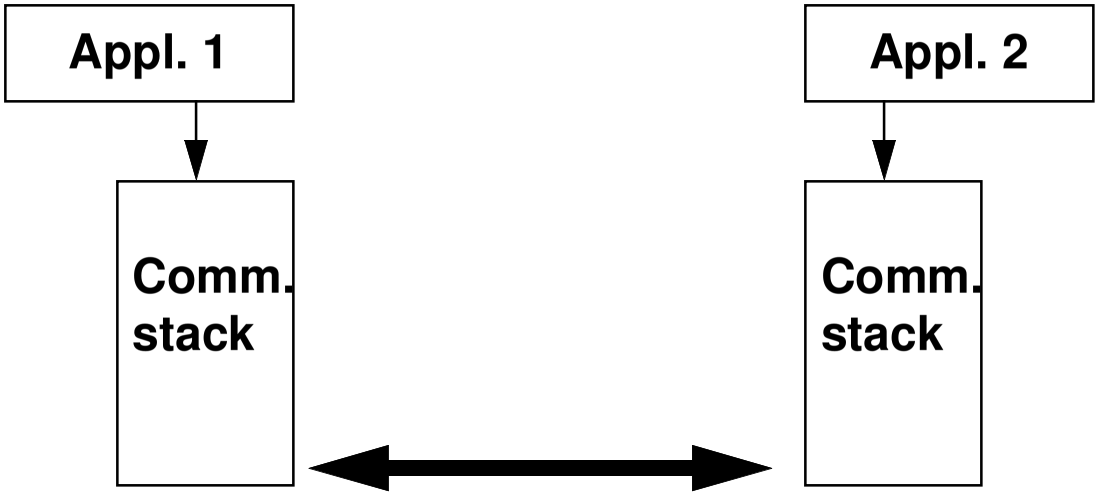
- **"API" vs. "Mechanism" vs. "Implementation"**
  - **an API is purely an interface specification**
    - » **defines what the services are**
    - » **defines calls and corresponding parameters**
    - » **typically accompanied by prog. language bindings**
    - » **commonly uses "tokens" (opaque binary buffers) for i/o**
  - **a mechanism is purely a specification**
    - » **defines how the services are accomplished**
    - » **defines optionality choices**
    - » **defines content and format of API tokens**
  - **an implementation is code**
    - » **conforms to API spec. and includes one or more mechs.**
    - » **typically packaged as a run-time library**

Introduction and Terminology (cont'd)

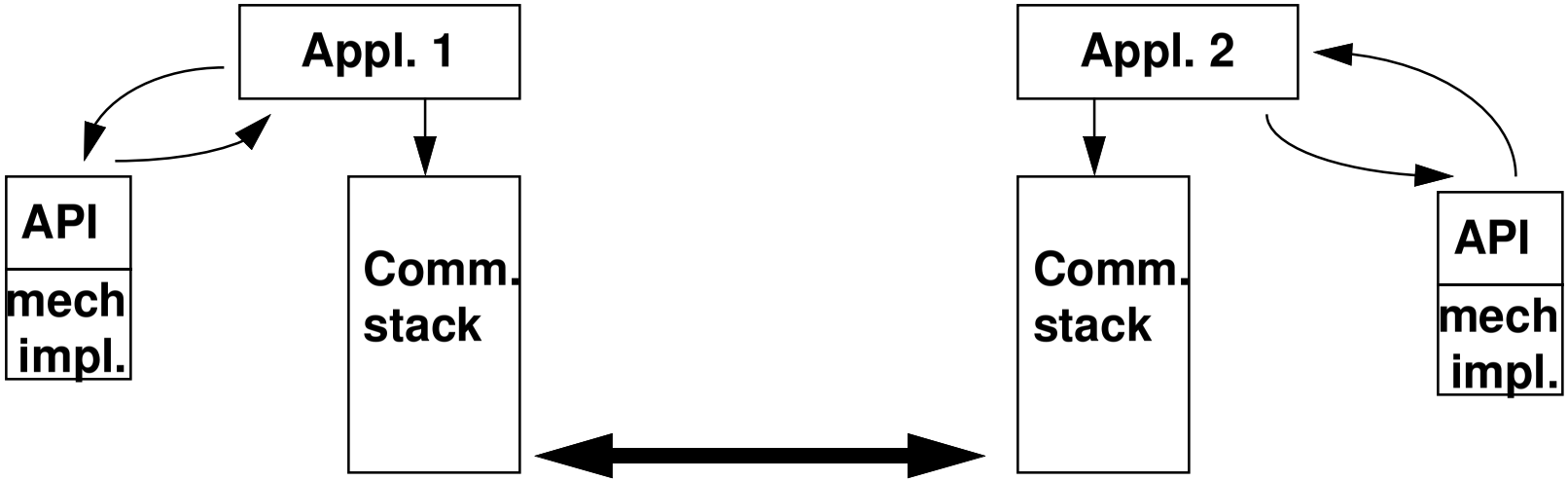
- **Portability Objectives**
  - **an application coding "generically" to an API should be portable across all implementations of all mechanisms of that API**
  - **an application which selects a particular mechanism should be portable across all implementations of that mechanism**

- **Data Transfer Independence**
  - **typically, APIs and mechanisms are independent of (application) data transfer**
  - **may send data for their own purposes, but this is transparent to calling application**
  - **assumption is that calling applications are themselves communicating entities**
  - ***API calls do not replace communications calls***

Communications Appl.



Application use of API



IETF CAT Group Activities

- **Two categories of communications requirements**
  - **on-line / session-oriented / real-time**
  - **off-line / e-mail-oriented / store-and-forward**
- **An objective is to provide public-key-based solutions for both categories**
- **Interesting "dichotomy"**
  - **session: API but few PK mechanism candidates**
  - **store-and-forward: PK mechanism candidates but no API**

IETF CAT Group Activities (cont'd)

- **Session:**
  - API is "Generic Security Service" (GSS) [RFCs 1508, 1509]
  - only candidate PK mechanism was DASS [RFC 1507]
    - not on standards track ("Experimental Protocol")
    - does not make full use of public-key infrastructure
- **Store-and-Forward:**
  - several candidate PK mechanisms ("enveloping protocols")
    - PEM [RFCs 1421-4], MOSS [RFC 1848]
    - S/MIME [RSA], MSP [DMS]
  - no API was defined
- **Need felt for Session mechanism and S-a-F API**

Outline

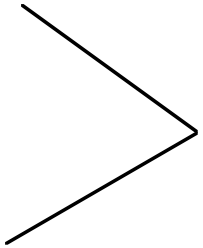
- Introduction and Terminology
- IETF CAT Group Activities
- ⇨ Session: GSS / SPKM
- Store&Forward: IDUP / PIM
- Comparisons with Other Work
- Conclusions

Session: GSS / SPKM

- **Generic Security Service (GSS)**
  - **an interface specification, along with C-language bindings**
  - **defines a set of calls for services which application developers can use to incorporate security**
  - **calls initiate, accept, and delete a "security context"**
    - » **secure (logical) connection between comm. entities**
    - » **data passed over this context can be integrity protected and encrypted with per-message calls**
  - **first GSS-API mechanism was based upon Kerberos v5**

GSS Calls

- **Acquire_credentials()**
- **Init_Sec_context() ; Accept_Sec_context()**
- **Sign()**
- **Seal()**



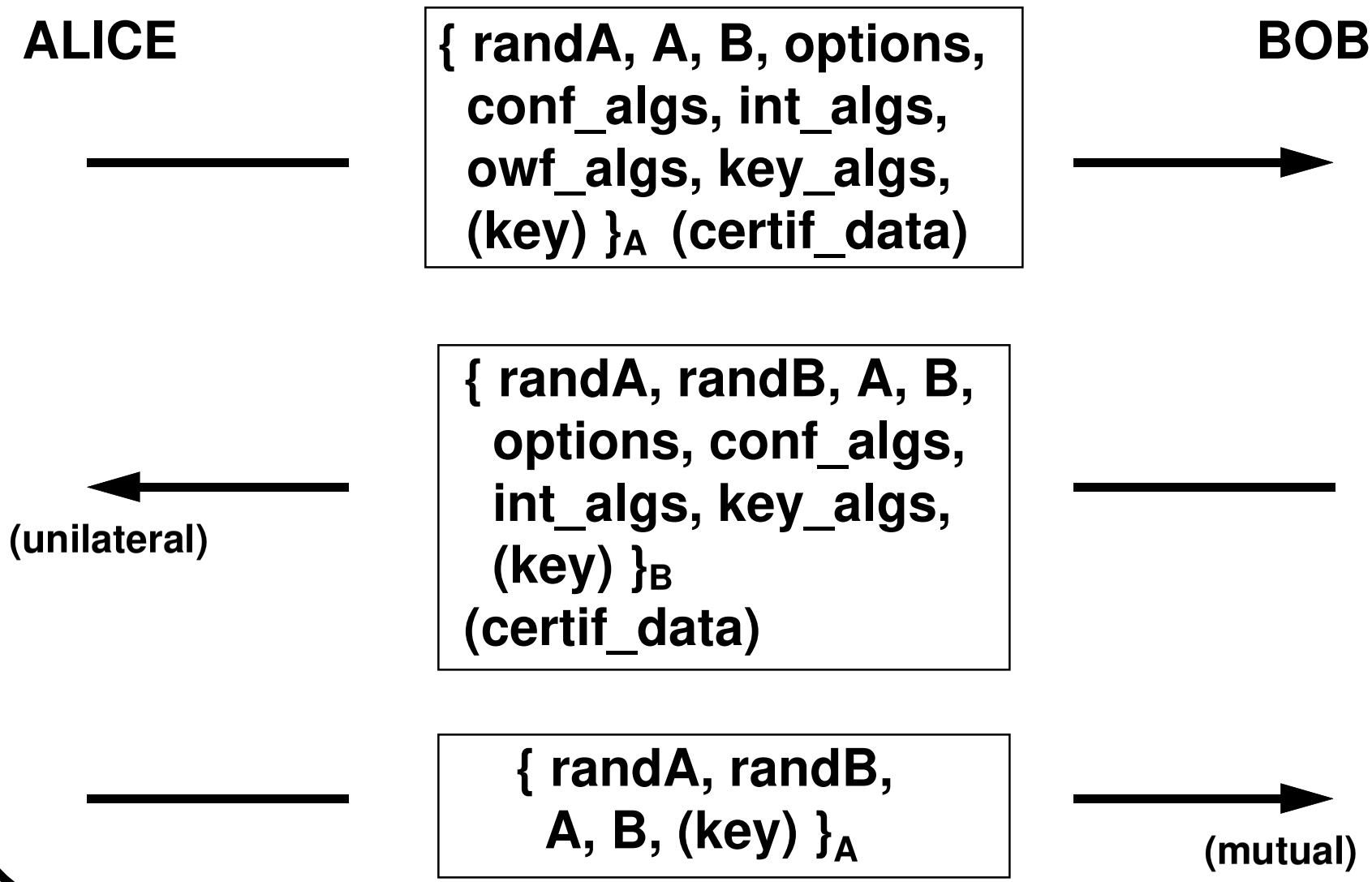
per message

- **Delete_Sec_context()**
- **Release_credentials()**

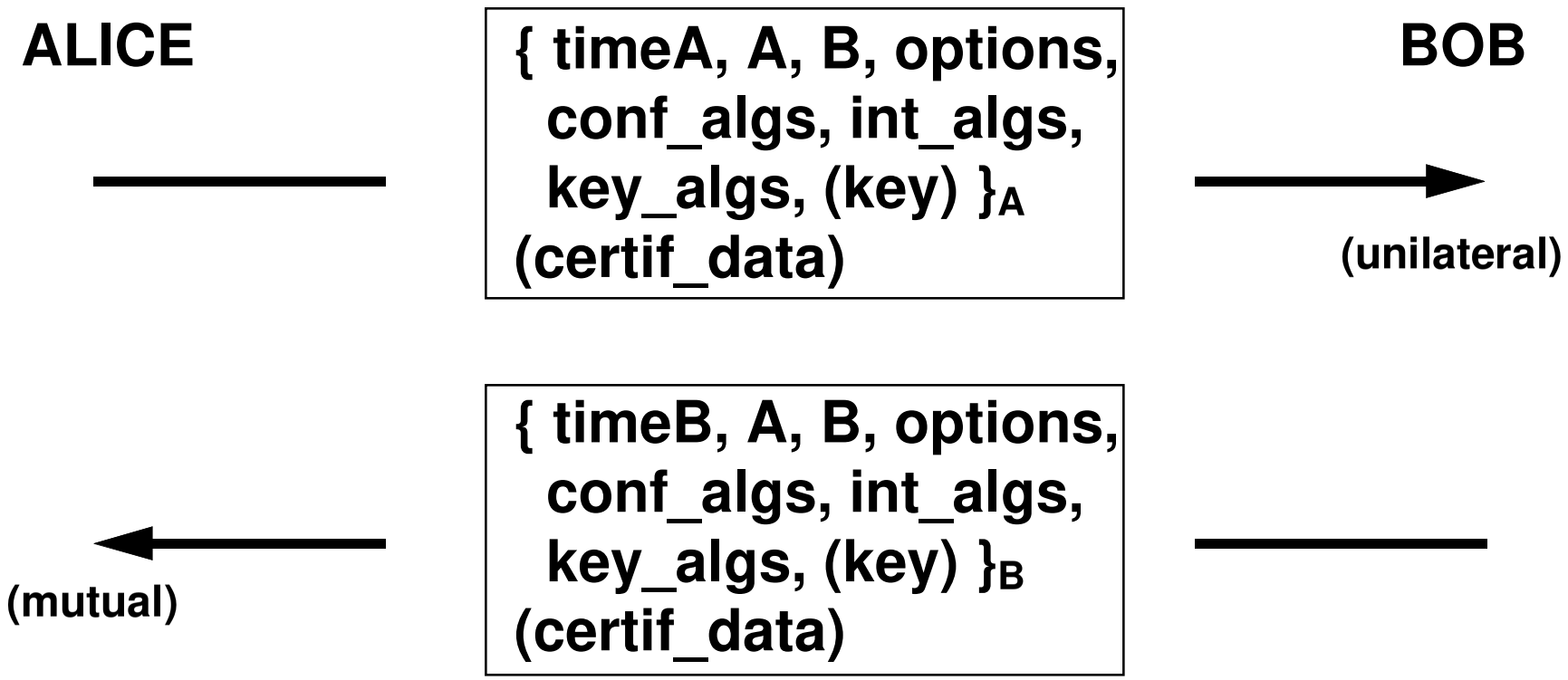
SPKM Feature Overview

- **"Simple Public-Key GSS-API Mechanism"**
- **At completion of context establishment:**
  - **all relevant certificate information exchanged**
  - **one side or both sides authenticated**
  - **negotiation of int., conf., key estb., and OWF algorithms**
  - **session key securely shared between comm. entities**
  - **options settled (e.g., sequencing, replay detection, lifetime)**
- **Over the context (per message, appl'n choice):**
  - **integrity with non-repudiation support (e.g., RSA)**
  - **integrity without non-repudiation support (e.g., DES-MAC)**
  - **confidentiality at "strong", "medium", or "weak" level**

SPKM-1 Context Estb. (Random Numbers)



SPKM-2 Context Estb. (Timestamps)



SPKM Feature Overview (cont'd)

- **Public Key Infrastructure for GSS calls**
- **Timestamp-based or random-number-based auth. (available to environments without secure time)**
- **Use of Alg. IDs wherever possible to allow maximum flexibility and growth**
- **Use of negotiation at context establishment to allow interoperability between environments**
  - **Key Estb., Confidentiality, Integrity, OWF Algorithms**
- **Use of generalized QOP structure (to select data integrity and confidentiality algs/services)**

SPKM Feature Overview (cont'd)

- **Based on Standards (Alg. IDs, X.509 Auth. Exchange)**
- **SPKM-conformant implementations can be exportable**
- **Can work with any underlying public-key infrastructure**
- **Per-message services need not use public-key operations (application choice)**
- **"Anonymous initiator" and "external / re-auth." authentication options supported**

Outline

- Introduction and Terminology
- IETF CAT Group Activities
- Session: GSS / SPKM
- ⇨ Store&Forward: IDUP / PIM
- Comparisons with Other Work
- Conclusions

S-a-F: IDUP / PIM

- **Objective is to define an API suitable for**
  - **PEM, MOSS, S/MIME, MSP, ...**
- **Flexible, general API should be of value to application developers**
  - **protection from mech. changes, new mechs.**
  - **no development delay waiting to see who "wins"**
  - **"security enveloping" can be done separately**
- **"Independent Data Unit Protection" (IDUP)**
  - **Extension of GSS-API for store-and-forward environment (i.e., re-use of as many calls as possible)**
- **"PEM-Based IDUP Mechanism" (PIM)**
  - **a mapping of RFC 1421 to the IDUP API calls**

IDUP Terminology / Concepts

- **"Protection"**
  - **protection operations yield a prot_token**
  - **IDU is either unchanged or changed (e.g., may be encoded or encrypted), but is referred to as "M-IDU"**
  - **prot_token (+ M-IDU) = "P-IDU"**
- **"Unprotection"**
  - **P-IDU is unprotected to decrypt and/or verify IDU**
  - **return codes give status of decrypt / verify operations**

IDUP Services

- **Three Types of Protection Services**
  - **perform unsolicited service (locally generated request)**
  - **perform solicited service (remotely generated request)**
  - **perform service solicitation (generate req. for remote end)**
- **Three Types of Unprotection Services**
  - **receive unsolicited service**
  - **receive solicited service**
  - **receive service solicitation**

Services (cont'd)

Originator (Protect)

perform unsolicited services (encrypt, sign)

perform service solicitation (request receipt)

Target (Unprotect)

receive unsolicited services (decrypt, verify)

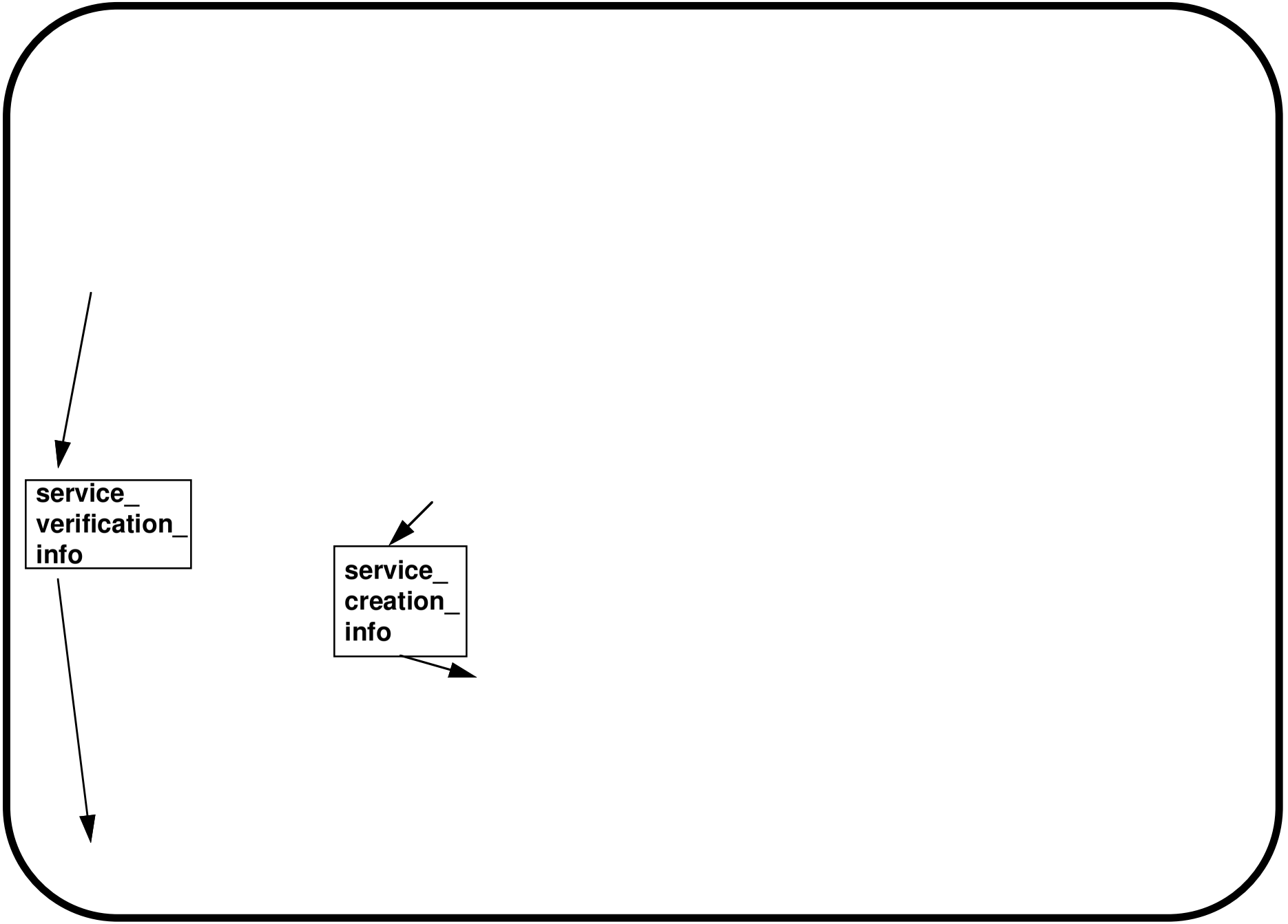
receive service solicitation (the receipt request)

Target (Protect)

perform solicited service (generate receipt)

Originator (Unprotect)

receive solicited service (verify receipt)


service_
verification_
info
service_
creation_
info

Services (cont'd)

- **IDUP-Defined Services (OBJ. IDs):**
  - **PERF_CONF (encrypt)**
  - **PERF_DOA (apply data origin authentication, integrity)**
  - **PERF_POO (apply non-repudiable proof of origin)**
  - **PERF_POD (apply non-repudiable proof of delivery)**

  - **REC_CONF (decrypt)**
  - **REC_DOA (verify data origin authentication, integrity)**
  - **REC_POO (verify proof of origin)**
  - **REC_POD (verify proof of delivery)**

  - **TTIME (use trusted time)**
  - **UTIME (use untrusted time)**

IDUP Function Calls

- **"Per-IDU" calls:**
  - **IDUP_Protect()**
  - **IDUP_Unprotect()**

  - **each call split into three parts**
    - » **Start_Operation() -- to identify target(s) / orig., QOP**
    - » **Operation() -- called once per buffer of input data (optional)**
    - » **End_Operation() -- called to return results (optional)**

IDUP Feature Overview

- **Ease of use for security services**
- **Flexibility to choose options and security services which match needs of app'n. or data**
- **General framework for generation / processing of "evidence" services**
- **Usability demonstrated by several mech. specs**
  - **PEM-based mechanism completed (PIM)**
  - **MSP-based mechanism also completed**
  - **S/MIME-based mechanism near completion**

Outline

- Introduction and Terminology
- IETF CAT Group Activities
- Session: GSS / SPKM
- Store&Forward: IDUP / PIM
- ⇨ Comparisons with Other Work
- Conclusions

Comparisons with Other Work

- **SPKM and other GSS-API mechanisms:**
  - **SPKM has true digital signatures, algorithm negotiation, non-reliance on secure time, initiator anonymity**
- **SPKM and Kerberos:**
  - **public-key vs. symmetric-key infrastructure**
  - **unprotected database vs. physically secure, on-line servers**
  - **“realm” ; “delegation” ; “ticket / TGT” ; “user creation”**
- **IDUP and other crypto. APIs:**
  - **different level (IDUP calls return true messaging objects)**
  - **IDUP can easily accommodate new types of security services**
  - **IDUP can be used by both “simple” and “sophisticated” (w.r.t. security) applications**

Conclusions

- **GSS / SPKM and IDUP / xIM**
  - **provide a set of necessary security services to application developers**
    - » **data origin authentication, data integrity, data confidentiality, evidence, support for non-repudiation**
  - **flexible, standards-based, easy to use**
- **Simple transition from “App.” to “Secure App.”**
  - **e.g., e-mail, DBMS, telnet**