Identity Based Key Management

Terence Spies
CTO
Voltage Security

Agenda

- The Key Management Problem
  - What it is, why customers want it, scenarios
- The ID-based Architecture
  - Symmetric keys, public keys
- What ID-based KM is good for
Scope

- Enterprise level key management for encryption in sensitive applications
- No answers here for:
  - Device keying (though this may improve that)
  - Signature, non-repudiation
- The main customer: enterprise developer that needs to encrypt data

Why?

- Encryption is painful and expensive
  - For users, this is evident
  - For developers, treated as a dangerous black art
- Do we need it to be otherwise?
  - Maybe not. One model is device keys everywhere.
    - Encrypt link to link, controller to media
  - But, probably yes.
    - App vulnerabilities
Why is this hard

The Function(s) We Need
A Taxonomy of Keys

- **Signature Creation**
  - (Cert Signing, Code Signing, S/MIME)
  - Replaceable
    - (regenerate, recertify)
  - Determinate
    - (public key sent to recipient)

- **Communication Privacy**
  - (VPN, SSL, IPSec)

- **Stored Data Privacy**
  - (S/MIME, WDE, Database)
  - Irreplaceable
    - (loss of key == loss of data)
  - Indeterminate
    - (recipient may be unbound)

Key Management Customers

**CISO**
- I need to be able to set policy
  - Who can access data
  - How they must authenticate
- I need to satisfy auditors
  - Logging and reporting
  - Key rotation requirements
- I need a strategy for future applications and regulations

**Enterprise Development**
- I need a straightforward way to encrypt application data
- I need to encrypt in complex environments with large datasets
- I need a way to secure data to
  - customers
  - partners / outsourcers
Key Management Scenarios

- Three archetypical application scenarios
  - Intranet portal document sharing
  - Customer communications
  - Partner-to-partner data sharing
- For each
  - Example architecture
  - Top-level requirements

Scenario 1 - Portal

- Company Q has an internal portal used to share documents. They wish to encrypt data so that documents will remain protected when copied off the portal
- All portal documents have associated ACLs containing group and individual identities
- The existing ACL should form the basis of encryption
Scenario 2 - Customers

- Agency A wants to be able to send sensitive information to a wide variety of citizens.
- Recipients may be specified by group.
- Recipients have minimal technical background, may reside on a variety of mail clients and Oses.

Scenario 3 - Partners

- Company X and Company Y wish to interchange documents securely via web and email.
- Each company only wants to authenticate it's own employees.
- Documents should be protected to groups and individuals.
Identity-Based Key Management

1. Map ID + key server parameter → key
2. Encrypt data
3. Request decryption key for ID
4. Calculate decryption key
5. Decrypt data

IBKM without IBE

1. Request ID + key server parameter → key
2. Encrypt data
3. Request decryption key for ID
4. Calculate decryption key
5. Decrypt data
Why IBKM?

- API / architecture that looks like access control
  - Encrypt(Data, SET OF Identities)
  - Decrypt(Data, Credential Name)
- Key name = ID, group, role name means free mapping to directory objects
  - RFC 822 names are as universal as they come
  - name@domain maps naturally into many models
  - Note that overlap with email is optional

IBKM Advantages

- Authentication at decryption time
  - Matches user expectations – authenticating to get information.
- Short-lived keys
  - Making a new public key is nearly free.
  - Time between identity binding and key issuance is short, as opposed to bind first systems.
- Binding to groups
  - ID = group or role, key server authenticates individual, checks membership.
What Doesn't It Do?

- **Authentication**
  - Relies on other, existing authentication
  - Does not communicate authentication policy
  - Encryptor relies on server policy
    - Encryptor chooses the key server
- **Signature**
  - IB signatures exist, but for the mainstream case, no operational difference....

---

**Scenario 1: Portal**

1. Map ACL element → key
2. Encrypt document
3. Authenticate via LDAP
4. Calculate decryption key
5. Decrypt document
Scenario 2: External Mail

1. Recipient address → key
2. Encrypt email (perhaps at gateway)
3. Authenticate via web page
4. Calculate decryption key
5. Decrypt email

Scenario 3: Partners

1. Map ACL element → key using partner params
2. Encrypt document
3. Authenticate via LDAP
4. Calculate decryption key
5. Decrypt document
Why IBKM is Important

- There is no good answer for secured communications outside the firewall to ad hoc recipients
- There is no good answer for group or role keying inside the enterprise
- IBKM provides a model for thinking about application level key management
- IBE provides an especially efficient way of implementing IBKM

For those viewing via webcast, please submit questions to kmwquestions@nist.gov