1. Signed emails: Clients’ actions

General actions:
- Certificate (public key) registration (between client and CA/RA)
- Certificate (public key) renewal (between client and CA)
- Certificate (public key) update (between client and CA)
- CRL update (from CA to repositories, periodically: once a day or week?)

Senders’ actions:
- Create email (email client locally)
- Create signature based on email contents (email client locally, local key-ring, certificate, and CPU involved)
- Send email together with the signature and certificate (extra network traffic involved: around 4K bytes)

Receivers’ actions:
- Retrieve emails (together with signatures and certificates) from SMTP server (extra network traffic involved)
- For each email, separate signature and certificate from the email
- Validate certificate
  - Retrieve CRL from repository (or local cache? Size=50M?)
  - Certification path construction
  - Certification path validation (local CPU)

2. Encrypted emails:

General actions:
- Certificate (public key) registration (between client and CA/RA)
- Certificate (public key) renewal (between client and CA)
- Certificate (public key) update (between client and CA)
- CRL update (from CA to repositories, periodically: once a day or week?)

Receivers’ actions:
- Send a signed email with his (encryption) certificate to the sender (go through all steps in 1.)
- Or, distribute his (encryption) certificate to a repository (say, a LDAP server)

Senders’ actions:
• Receive the receiver’s signed email with signature and certificates (from SMTP server)
• Or, discover and retrieve the receiver’s certificate directly from a repository
• Validate the certificate
  o Retrieve CRL from repository (or local cache? Size=50M?)
  o Certification path construction
  o Certification path validation (local CPU)
• If valid, store it to local key-ring or address book
• Create a new email and encrypt it with the receiver’s public key
• Send the encrypted email

3. Notes:

(1) Since the major concern about scalability is on the repository side (for example, CRL download time, certificates retrieval for path construction, etc.), we need to focus on modeling repository and its related transactions and metrics (say, network bandwidth consumption, delay, etc.).
(2) Packet-based or statistics-based simulation? Still confused…
(3) Storage and CPU issue at a repository?