Secure Emails

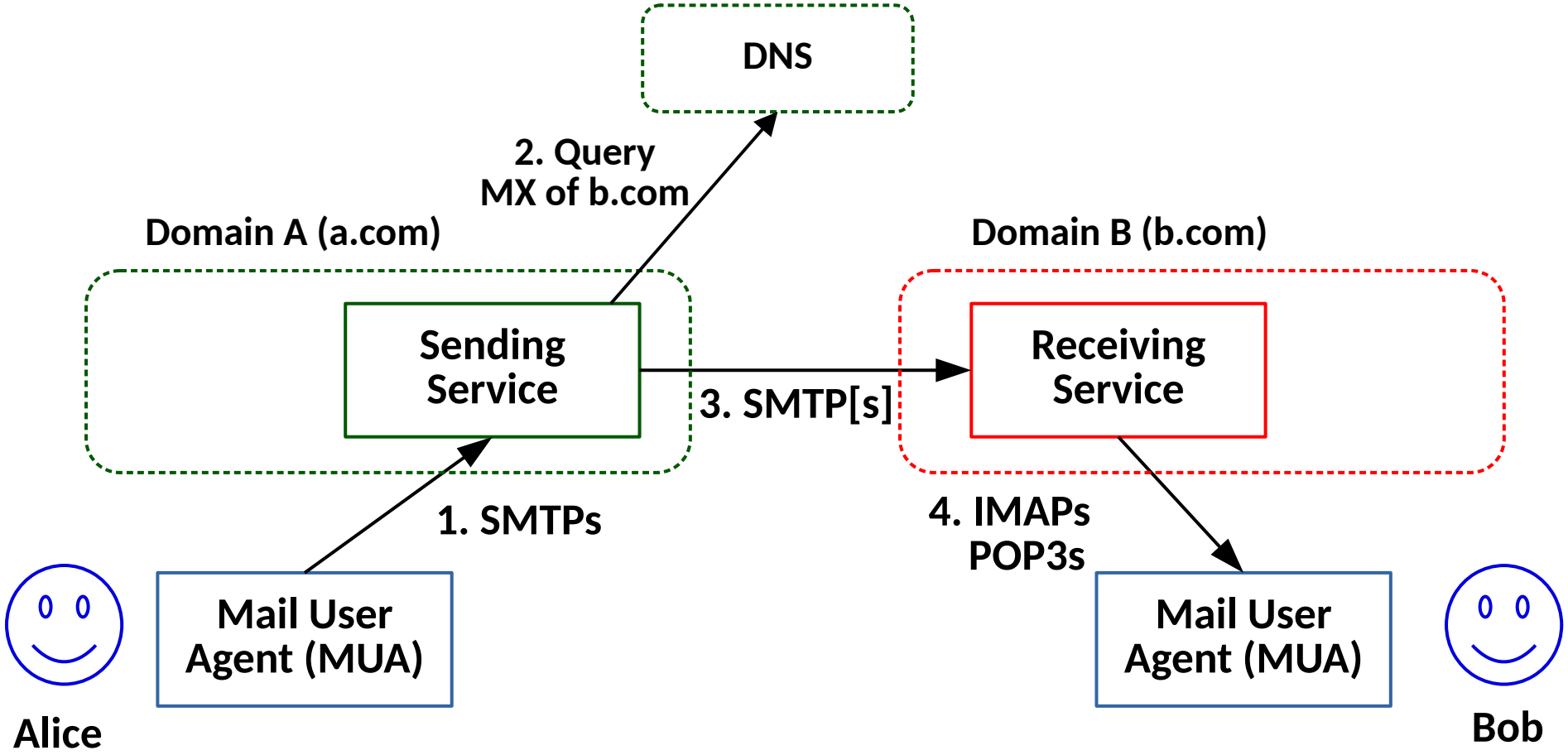
Luca Ferretti

Protocolli e Architetture Email

Laurea Informatica

Università degli Studi di Modena e Reggio Emilia

Email Architecture



We assume protocols **operated over secure channels** (over TLS)

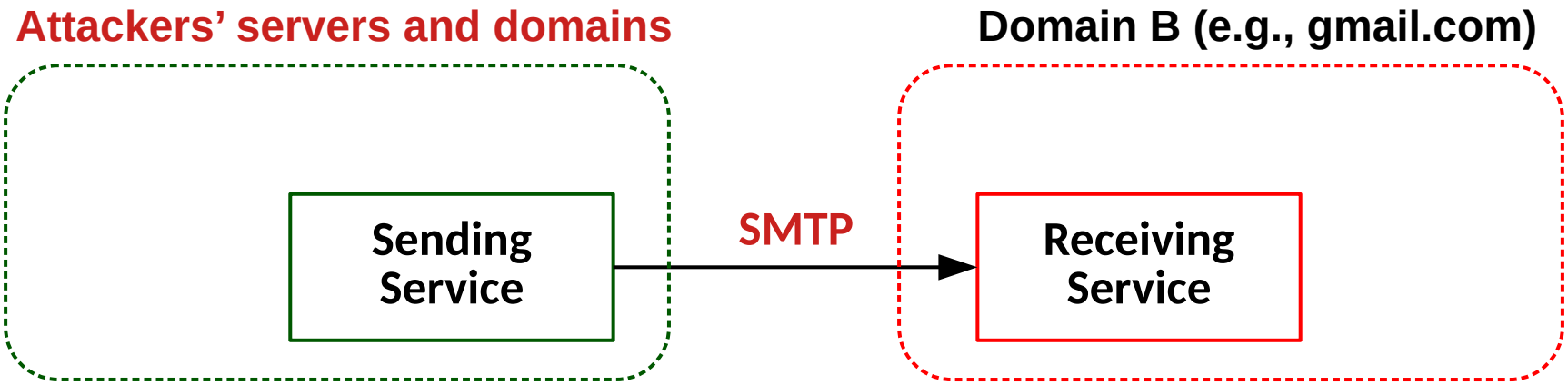
- Thus, we already defend against “external” attackers

Spoofing email addresses [1]

- **Spoofing** is an attack where the attacker is able to disguise the sender of a message as a legitimate source
  - We discussed that sender authenticity can be guaranteed by using cryptographic protocols (digital signatures, MACs)
  - However, many protocols do not use cryptographic schemes
    - Legacy protocols that were not designed for security
    - Protocols constraints
- Protecting against spoofing in emails has always been quite complex due its distributed design
  - A large number of email providers exist, each with a large number of users
  - Each provider might know its own users, but it cannot know the identities (and credentials) of other providers

Spoofing email addresses [2]

- Emails do not have any end-to-end security guarantees
  - Service providers can read our email in plaintext
- The SMTP protocol does not have authentication mechanisms to propagate the identity of the user
  - The identity of the user can be arbitrarily declared within **the headers of the email**
    - `MAIL FROM` or `From` headers
    - An attacker can send an email from its own domain and spoof another the identity of another user

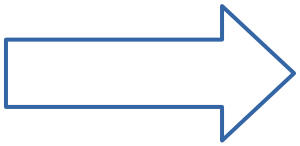


Spoofing email addresses in emails [3]

- The identity of the user can be arbitrarily declared within **the headers of the email**

SMTP message

```
HELO a.com
MAIL FROM: alice@a.com
RCTP TO: bob@b.com
```



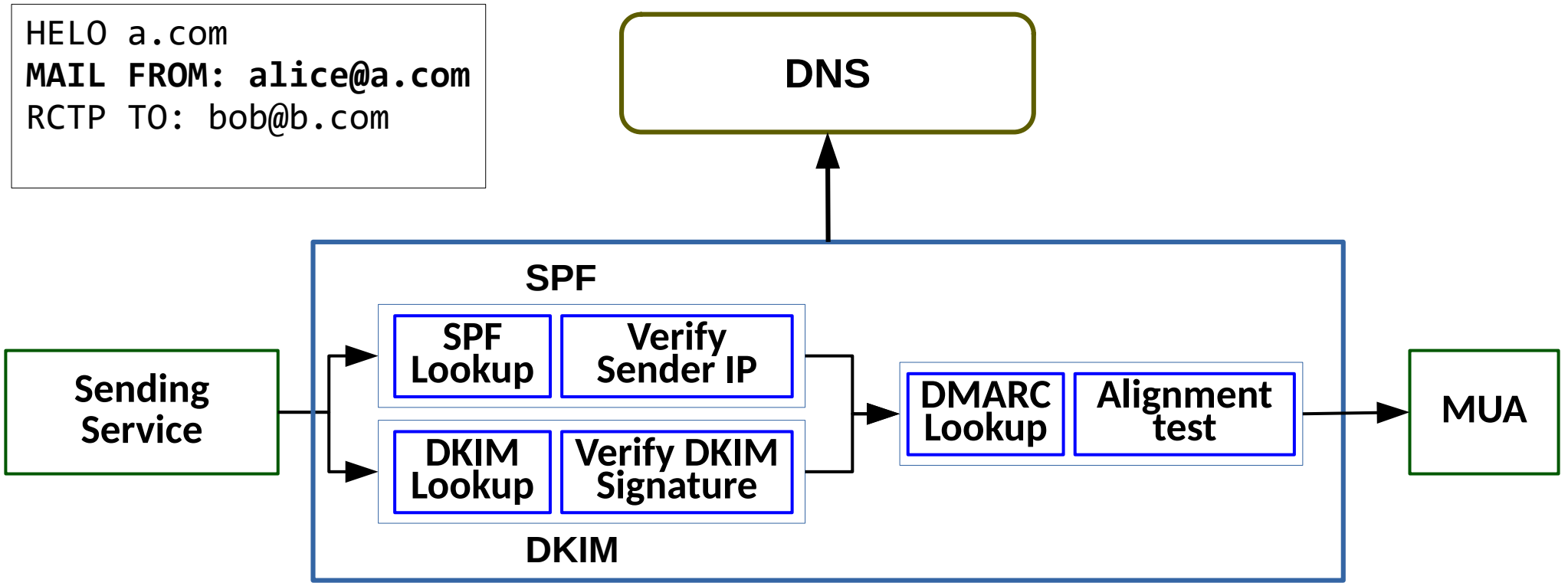
MUA Message Header

```
From: <alice@a.com>
To: <bob@b.com>
Subject: Hello, World!
```

Attaching security to emails

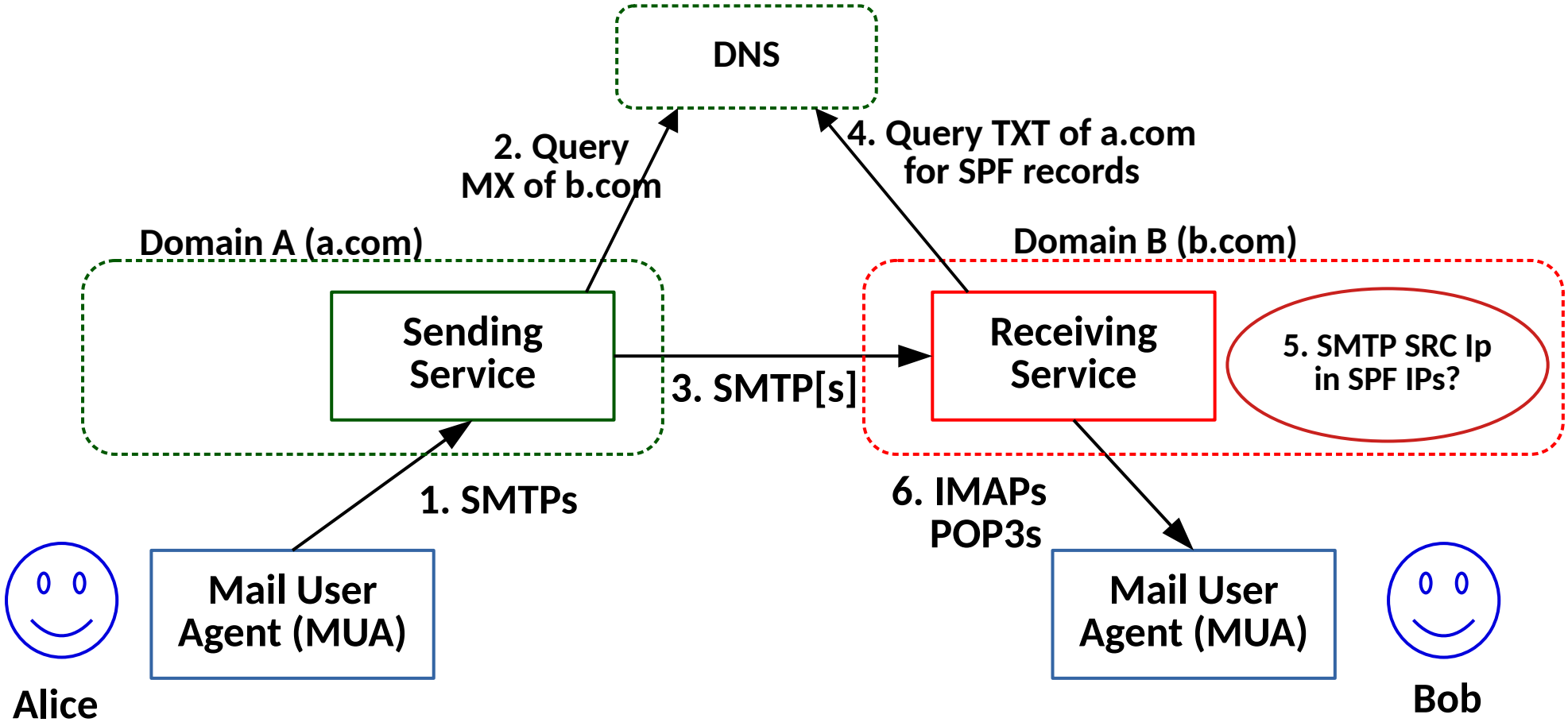
- Security measures at the providers' side to defend against sender spoofing
  - **Sender Policy Framework (SPF)**
  - **DomainKeys Identified Mail (DKIM)**
  - **Domain-based Message Authentication, Reporting & Conformance (DMARC)**
  - **Brand Indicators for Mes-sage Identification (BIMI)**
  - **Authenticated Received Chain (ARC)**
- End-to-end approaches to protect confidentiality and authenticity (users' side):
  - **S/MIME**
  - **PGP**
- Email security and legal value in Italy: **PEC**

Email server-side security measures



- **SPF** requires the domain of an Email provider (**a.com** in the example) to publish the list of the **authorized sending services (DNS names, IPs)** in its DNS TXT records
- **DKIM** has a similar aim, but let the sending service sign email headers. **DNS TXT records include public keys** to allow verifying signatures
- **DMARC** is a further check to adopt both SPF and DKIM
- Other protocols (ARC, BIMI) might be adopted for similar and other checks
- However, attacks might still be possible due to specific implementation choices
  - see recent attacks

Security Policy Framework (SPF)



SPF IPs in TXT DNS records define allowed source IP addresses for a domain

- See some additional details e.g. https://www.cloudflare.com/learning/dns/dns-records/dns-spf-record/

End-to-end security (extra)

End-to-end email protection

- Servers' side security solutions might be useful to massively protect the email system, however they **are not end-to-end** and have limitations
- **They are not mandatory, it is the choice of the receiver provider to implement them**
  - **Users must rely on their own provider to implement and configure them correctly**
  - These security solutions are **not mandatory for senders**, but good receivers consider senders that do not use them as **unreliable**
    - **Unreliable senders could cause emails to be classified as SPAM**
- They might not be perfect, and some spoofing attack might still be able to succeed (although very difficult when using good email providers)
- They do not protect against attacks against email servers themselves
  - And do not protect email end-to-end confidentiality

End-to-end security for emails

- Improved solutions require **end-to-end security** from user to user
- Similar approaches, although technically different:
  - **S/MIME**: integrate PKI and x509 certificates with emails
  - PGP: encryption and signature guarantees, historical open source project and implementation for strong encrypted emails

S/MIME

- **S/MIME**: integrate PKI and x509 certificates with emails
  - Standardized in 1995 by RSA Data Security Inc.
  - Now Standard IETF
- S/MIME refers to **Secure MIME**, that is used to actually sign and (optionally) encrypt messages
  - MIME is the standard defining encoding techniques to transmit any type of data format through email (that is a text-based protocol)
  - S/MIME defines how to encapsulate standard MIME data into standard signature and encryption scheme
- Emails also attach **x509 certificates** and **certificate chains**
  - Email clients maintain root Certification Authorities to verify certificates as browsers typically do for Web certificates
  - Certificates can be purchased at some CA
  - Can also deploy a private hierarchy as discussed for Web servers

Free S/MIME Certificates

- There is no LetsEncrypt-like project that officially releases free certificates automatically, but there may exist certification authorities that free services
  - https://extrassl.actalis.it/portal/uapub/freemail?lang=en
  - Sends you a test code at the email that must be verified, then sends you the certificate and the private key (encrypted with a password
    - In this case, the certificates are encoded by using the PKCS12 standard with **`pfx`** estension (different from PEM). Typically, email clients support it natively, but can also be opened with openssl by using the following command:

      `openssl pkcs12 -in <pfx-file>`
  - Note: similar to LetsEncrypt, it has many limitations, and if the attacker compromises your email and obtains a valid certificate in the meanwhile, he can also send legitimately signed emails

Pretty Good Privacy

- PGP is an historical cryptographic protocol for signing and encrypting data
  - data → asynchronous communications
  - hybrid asymmetric data encryption (symmetric key wrapping)
- PGP is based on a combination of symmetric crypto, hash functions, asymmetric crypto and digital signatures
  - each “block” might be implemented through different protocols (e.g., RSA *vs.* DSA for digital signatures)

OpenPGP

- OpenPGP is the de-facto standard framework for asynchronous communications
  - PGP is the underlying cryptographic protocol
- Decentralized system
  - no certification authorities
  - Web of trust
- Gnu Privacy Guard (GPG) is a free implementation of the OpenPGP framework
  - Can use it with the **gpg** or **gpg2** command (gpg2 is the suggested version, sometimes already replaces gpg as the default and the two command are aliases)

Key pairs

- Each user owns a **Key pair**
  - *secret key*
  - *public key*

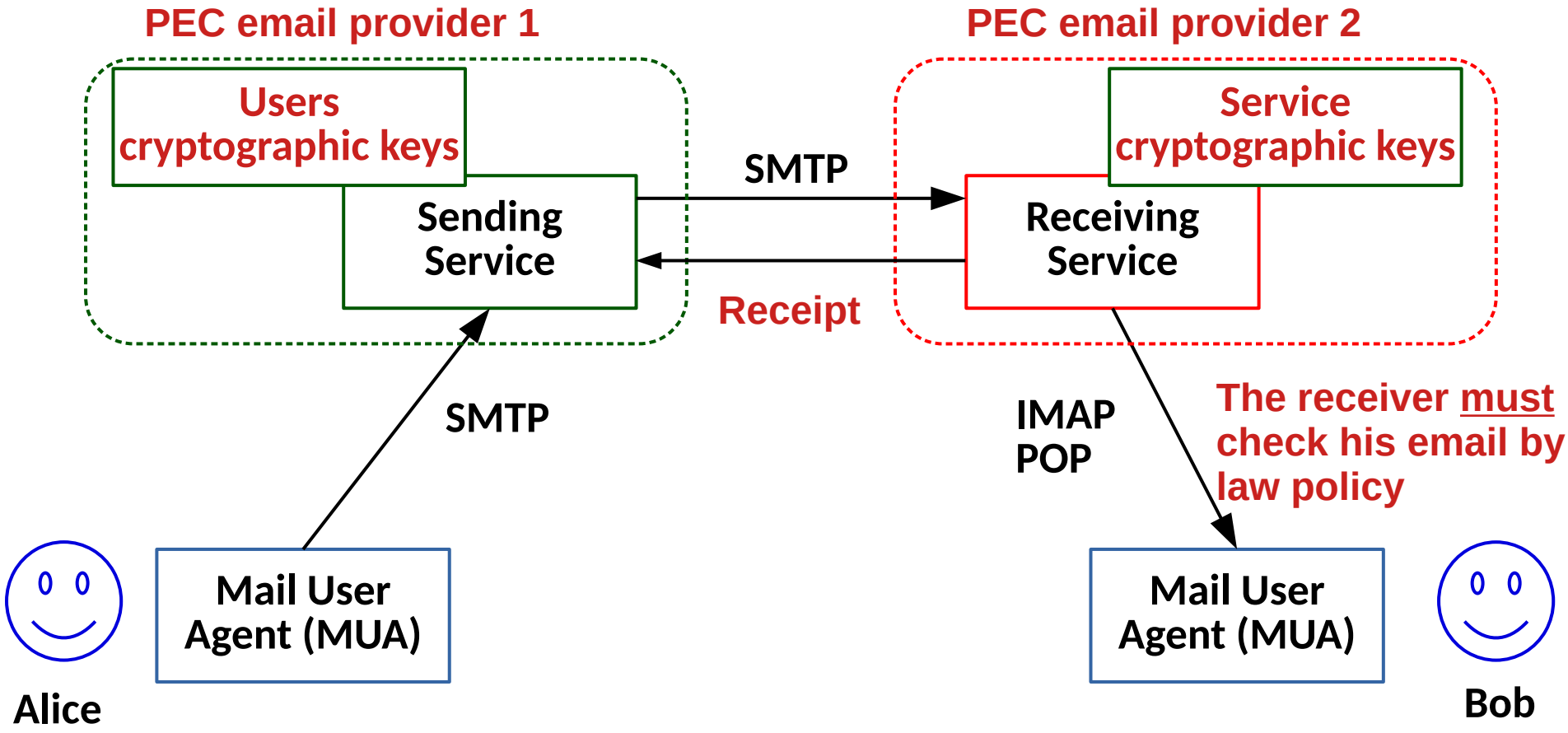
- *When used to send asynchronous messages to other users, the keys serve as for any similar protocol (e.g., S/MIME)*
  - *the secret key allows the user to **sign** messages*
  - *the public key allows others to encrypt data for the user*

- *When used to encrypt data (e.g., backups) the user can use both keys to **encrypt and sign data for himself***

PEC (Posta Elettronica Certifica)

- Tecnicamente, realizzato tramite tecnologia S/MIME
  - Le CA root impiegate sono aziende pubbliche o private autorizzate allo scopo
- Sfrutta le garanzie di **non repudiabilità** delle firme digitali per dare caratteristiche di valore giuridico alle email
  - Giuridicamente equivalente a una raccomandata con ricevuta di ritorno
  - Esiste ed è valida solo in Italia
  - **Non supporta la cifratura delle informazioni**
- Definisce in modo più stringente diversi aspetti tecnologi (e.g., protocolli supportati, standard di codifica impiegati)
  - Ha un suo RFC → RFC6109 - Italian Certified Electronic Mail

PEC: garanzie di ricezione?



- The sender service (on behalf of the user) signs the email
- The receiving service sends back a timestamped and signed receipt

  → **Both sending and receiving events are non-repudiable**

  → **The user is demanded to check his email by law**