b
UNIVERSITÄT
BERN

b
UNIVERSITÄT
BERN

# Network Security

# XI. Electronic Mail and Domain Name System

**Prof. Dr. Torsten Braun, Institut für Informatik**

Bern, 09.05.2022 – 16.05.2022

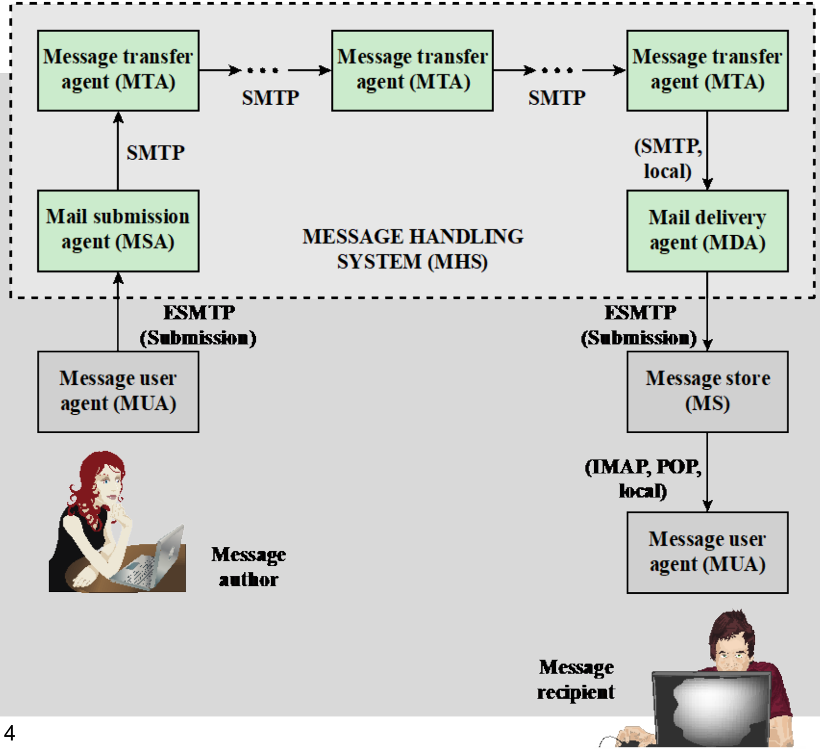
# Electronic Mail

## Table of Contents

1. Internet Mail Architecture
2. Email Formats
3. Email Threats
4. Pretty Good Privacy
5. Secure/Multipurpose Internet Mail Extensions
6. Domain Name System Security
7. DNS-based Authentication of Named Entities
8. Sender Policy Framework
9. Domain Keys Identified Mail
10. Domain-based Message Authentication, Reporting, and Conformance

# 1. Internet Mail Architecture

## 1. Email Protocols and Modules



- Message User Agent
  - operates on behalf of user actors and user applications.
  - formats message and performs initial submission into MHS via MSA.
  - processes received mail for storage and/or display to the recipient user.
- Mail Submission Agent
  - accepts message submitted by MUA.
  - enforces policies of hosting domain.
- Message Transfer Agent
  - relays mail for one application-level hop.
  - adds trace information to message header.
- Mail Delivery Agent
  - transfers message from Message Handling System to Message Store.
- Message Store
  - An MUA can employ a long-term MS.
  - MS can be located on a remote server or on same machine as MUA.
  - Typically, MUA retrieves messages from a remote server using Post Office Protocol or Internet Message Access Protocol.

# 1. Internet Mail Architecture

## 2.1 Simple Mail Transfer Protocol

- Direct TCP connections between servers
- Transmission of multiple messages over a TCP connection in both directions
- Messages
  - ASCII text only
  - Maximum message length < 64 KB in older implementations
  - Message formats
    - RFC822 (Text)
    - Multipurpose Internet Mail Extensions

Commands between client and server

- HELO: introduction
- MAIL FROM: sender
- RCPT TO: receiver
- DATA: message data
- QUIT: end

# 1. Internet Mail Architecture

## 2.2 SMTP Operation

```
% telnet asterix 25
Trying 130.92.64.4...
Connected to asterix.
Escape character is '^]'.
220 asterix.iam.unibe.ch Sendmail SMI-8.6/SMI-SVR4 ready at Fri, 20 Feb 1998 12:40:16
   +0100
HELO iam.unibe.ch
250 asterix.iam.unibe.ch Hello akela [130.92.65.44], pleased to meet you
MAIL FROM:<braun>
250 <braun>... Sender ok
RCPT TO:<braun>
250 <braun>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Hallo Torsten.
.
250 MAA00591 Message accepted for delivery
QUIT
221 asterix.iam.unibe.ch closing connection
Connection closed by foreign host.
%
```

# 1. Internet Mail Architecture

## 3. Client Protocols

### Post Office Protocol

- allows an email client to download an email from an email server.
- POP3 UA connects to server via TCP, TCP port 110
- After authorization, UA can issue POP3 commands to retrieve and delete mail.

### Internet Mail Access Protocol

- enables email client to access email on email server.
- TCP port 143
- more complex than POP3
- provides stronger authentication and provides other functions not supported by POP3.

# 2. Email Formats

## 1.1 RFC 822/5322 Format

```
Message-ID:
<34EDE48C.FF554EC5@iam.unibe.ch>

Date: Fri, 20 Feb 1998 21:16:13
+0100

From: Torsten Braun
<braun@iam.unibe.ch>

To: t.braun@ieee.org

Subject: Hallo

Cc: t.braun@acm.org
```

Header

Body

```
Hello Torsten.
```

More header fields

- **`Bcc:`**
- **`Reply-To:`**
- **`In-Reply-To:`**

# 2. Email Formats

## 1.2 SMTP/RFC 5322 Limitations

**SMTP**

- cannot transmit executable files or other binary objects.
- cannot transmit text data that includes national language characters.
- servers may reject mail message over a certain size.

# 2. Email Formats

## 2.1 MIME Specifications

- New message header fields
- Definition of content formats standardizing representations that support multimedia email
- Transfer encodings enable conversion of any content format into a form that is protected from alteration by the mail system

# 2. Email Formats

## 2.2 MIME-Mail

```
Message-ID:
<34EDF402.FB7DA3B9@iam.unibe.ch>
Date: Fri, 20 Feb 1998 22:22:10 +0100
From: Torsten Braun <braun@iam.unibe.ch>
MIME-Version: 1.0
To: t.braun@ieee.org
Subject: MIME Beispiel
Content-Type: multipart/mixed; boundary=
"------------5EF727B907217426DF5C4EE0"
This is a multipart message in mime
format.
```

```
--------------5EF727B907217426DF5C4EE0
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: 7bit
<HTML>
<B><FONT FACE="Arial">Hello
   Torsten.</FONT></B>
<BR>&nbsp;</HTML>
--------------5EF727B907217426DF5C4EE0
Content-Type: text/plain; charset=us-
   ascii; name="test.txt"
Content-Transfer-Encoding: 7bit
Content-Disposition: inline;
   filename="test.txt"
Testdatei
--------------5EF727B907217426DF5C4EE0--
```

# 2. Email Formats

## 2.3 MIME Header Fields

mandatory

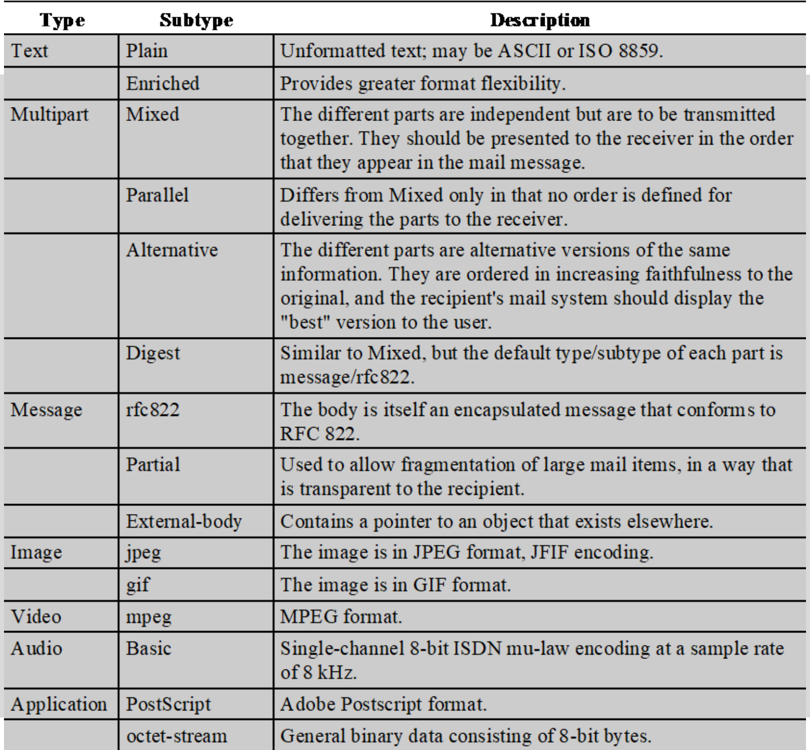
- **MIME-Version**: 1.0: indicates that message conforms to RFCs 2045/2046.
- **Content-Type** describes data contained in the body with sufficient detail that the receiving user agent can select an appropriate agent or mechanism to represent the data to the user
- **Content-Transfer-Encoding** indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport.

optional

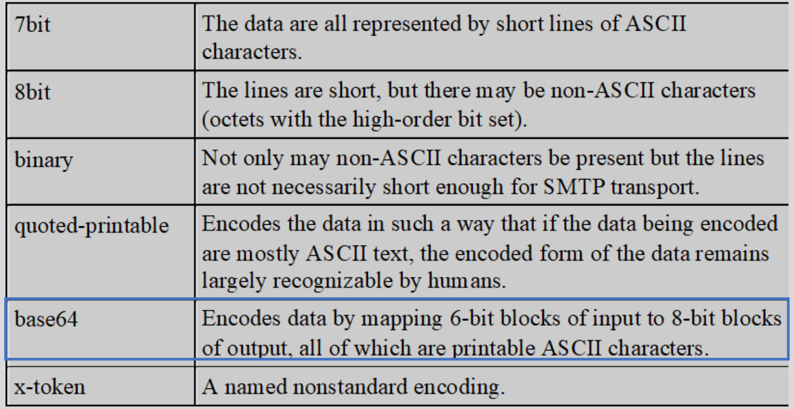
- **Content-ID** identifies MIME entities uniquely in multiple contexts.
- **Content-Description** is a text description of the object with the body; this is useful when the object is not readable (e.g., audio data).

# 2. Email Formats

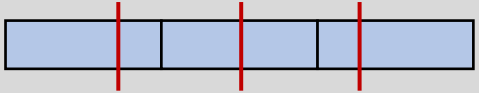
## 2.4 MIME Content Types and Transfer Encodings



| Type | Subtype | Description |
|---|---|---|
| Text | Plain | Unformatted text; may be ASCII or ISO 8859. |
| | Enriched | Provides greater format flexibility. |
| Multipart | Mixed | The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message. |
| | Parallel | Differs from Mixed only in that no order is defined for delivering the parts to the receiver. |
| | Alternative | The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original, and the recipient's mail system should display the "best" version to the user. |
| | Digest | Similar to Mixed, but the default type/subtype of each part is message/rfc822. |
| Message | rfc822 | The body is itself an encapsulated message that conforms to RFC 822. |
| | Partial | Used to allow fragmentation of large mail items, in a way that is transparent to the recipient. |
| | External-body | Contains a pointer to an object that exists elsewhere. |
| Image | jpeg | The image is in JPEG format, JFIF encoding. |
| | gif | The image is in GIF format. |
| Video | mpeg | MPEG format. |
| Audio | Basic | Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8 kHz. |
| Application | PostScript | Adobe Postscript format. |
| | octet-stream | General binary data consisting of 8-bit bytes. |



| | |
|---|---|
| 7bit | The data are all represented by short lines of ASCII characters. |
| 8bit | The lines are short, but there may be non-ASCII characters (octets with the high-order bit set). |
| binary | Not only may non-ASCII characters be present but the lines are not necessarily short enough for SMTP transport. |
| quoted-printable | Encodes the data in such a way that if the data being encoded are mostly ASCII text, the encoded form of the data remains largely recognizable by humans. |
| base64 | Encodes data by mapping 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters. |
| x-token | A named nonstandard encoding. |



# 3. Email Threats

## 1. Classification

- Authenticity
- Integrity
- Confidentiality
- Availability

related

# 3. Email Threats

## 2.1 Email Threats and Mitigations

| Threat | Impact on Purported Sender | Impact on Receiver | Mitigation |
|---|---|---|---|
| Email sent by unauthorized MTA in enterprise,<br>e.g., malware botnet | Loss of reputation, valid email from enterprise may be blocked as possible spam/phishing attack. | Unsolicited Bulk Email and/or email containing malicious links may be delivered into user inboxes | Deployment of domain-based authentication techniques.<br>Use of digital signatures over email. |
| Email message sent using spoofed or unregistered sending domain | | | |
| Email message sent using forged sending address or email address,<br>i.e., phishing, spear phishing | | UBE and/or email containing malicious links may be delivered.<br>Users may inadvertently divulge sensitive information or PII. | |

# 3. Email Threats

## 2.2 Email Threats and Mitigations

| Threat | Impact on Purported Sender | Impact on Receiver | Mitigation |
|---|---|---|---|
| Email modified in transit | Leak of sensitive information or Personally Identifiable Information. | Leak of sensitive information, altered message may contain malicious information | Use of TLS to encrypt email transfer between server.<br>Use of end-to-end email encryption. |
| Disclosure of sensitive information (e.g., PII) via monitoring and capturing of email traffic | | | |
| UBE (i.e., spam) | None, unless purported sender is spoofed. | UBE and/or email containing malicious links may be delivered into user inboxes | Techniques to address UBE. |
| DoS/DDoS attack against an enterprises' email servers | Inability to send email. | Inability to receive email. | Multiple mail servers, use of cloud-based email providers. |

# 3. Email Threats

## 3.1 Counter Threat Protocols

- STARTTLS
  - SMTP security extension that provides authentication, integrity, non-repudiation, confidentiality for the entire SMTP message by running SMTP over TLS
- S/MIME
  - provides authentication, integrity, non-repudiation, confidentiality of the SMTP message body.
- DNS Security Extensions
  - provides authentication and integrity protection of DNS data.
- DNS-based Authentication of Named Entities
  - overcomes problems in the Certificate Authority system by providing an alternative channel for authenticating public keys based on DNSSEC.

# 3. Email Threats

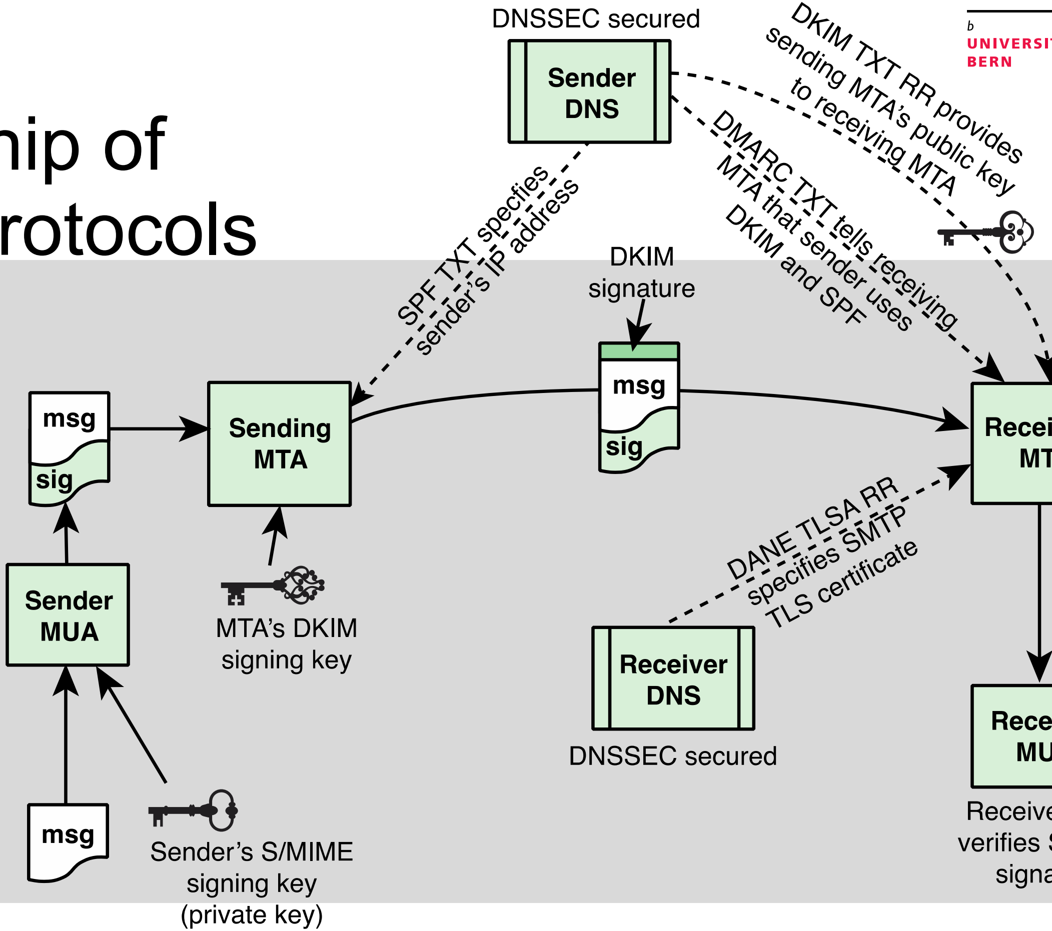
## 3.2 Counter Threat Protocols

- Sender Policy Framework
  - uses DNS to allow domain owners to create records that associate the domain name with a specific IP address range of authorized message senders.
  - Receivers check the SPF TXT record in DNS to confirm that the purported sender of a message is permitted to use that source IP address and reject mail that does not come from an authorized IP address.

- Domain Keys Identified Mail
  - enables an MTA to sign selected headers and body of a message.
  - validates the source domain of the mail and provides message body integrity.
- Domain-based Message Authentication, Reporting, and Conformance
  - informs senders about the proportionate effectiveness of their SPF and DKIM policies
  - signals to receivers what action should be taken in various individual and bulk attack scenarios

# 3. Email-Threats

## 3.3 Interrelationship of Counter Threat Protocols


DNSSEC secured
Sender DNS
DKIM TXT RR provides sending MTA's public key to receiving MTA
DMARC TXT tells receiving MTA that sender uses DKIM and SPF
SPF TXT specifies sender's IP address
DKIM signature
msg
sig
Sending MTA
Receiving MTA
Sender MUA
MTA's DKIM signing key
DANE TLSA RR specifies SMTP TLS certificate
Receiver DNS
DNSSEC secured
Receiver MUA
Receiver MUA verifies S/MIME signature
Sender's S/MIME signing key (private key)

# 4. Pretty Good Privacy

## Example Email

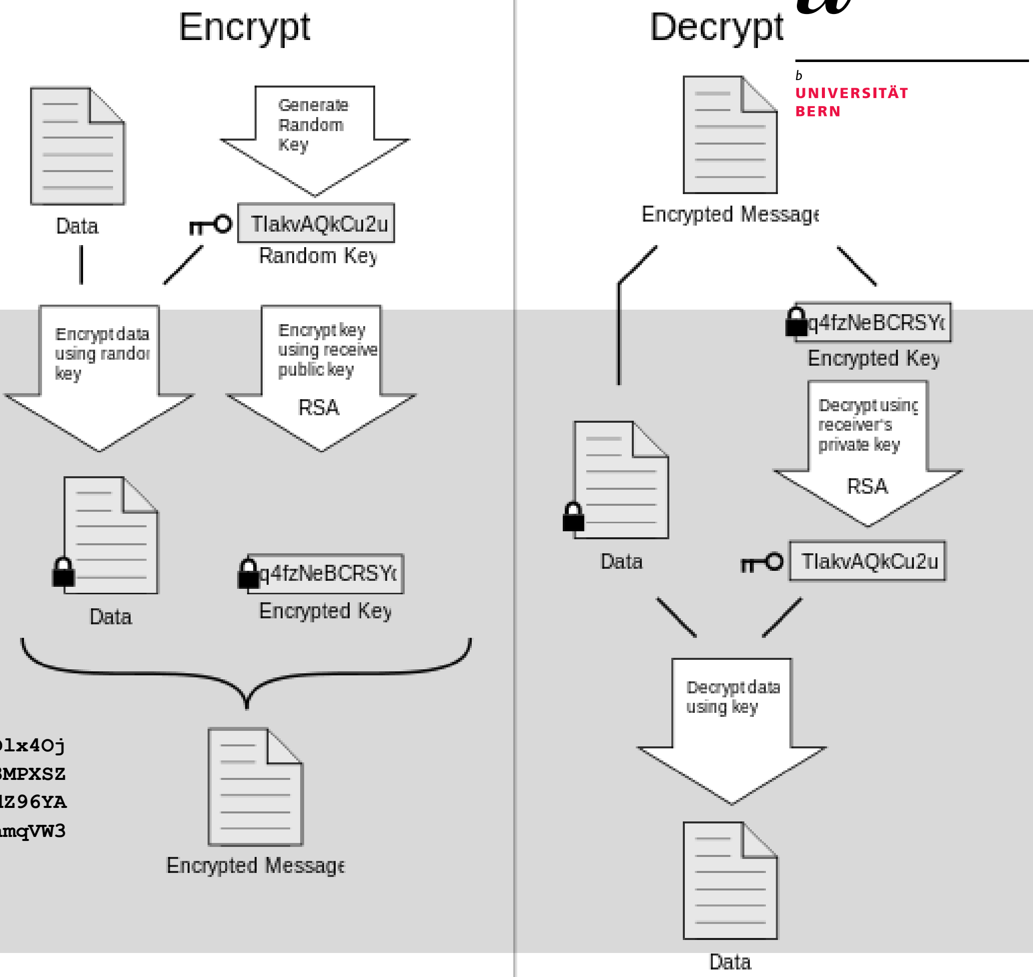
```
From: Michael Elkins <elkins@aero.org>
To: Michael Elkins <elkins@aero.org>
Mime-Version: 1.0
Content-Type: multipart/encrypted; boundary=foo;
        protocol="application/pgp-encrypted"

--foo
Content-Type: application/pgp-encrypted
Version: 1

--foo
Content-Type: application/octet-stream

-----BEGIN PGP MESSAGE-----
Version: 2.6.2
hIwDY32hYGCE8MkBA/wOu7d45aUxF4Q0RKJprD3v5Z9K1YcRJ2fve87lMlDlx4Oj
eW4GDdBfLbJE7VUpp13N19GL8e/AqbyyjHH4aS0YoTk10QQ9nnRvjY8nZL3MPXSZ
g9VGQxFeGqzykzmykU6A26MSMexR4ApeeON6xzZWfo+0yOqAq6lb46wsvldZ96YA
AABH78hyX7YX4uT1tNCWEIIBoqqvCeIMpp7UQ2IzBrXg6GtukS8NxbukLeamqVW3
1yt21DYOjuLzcMNe/JNsD9vDVCvOOG3OCi8=
=zzaA
-----END PGP MESSAGE-----

--foo--
```



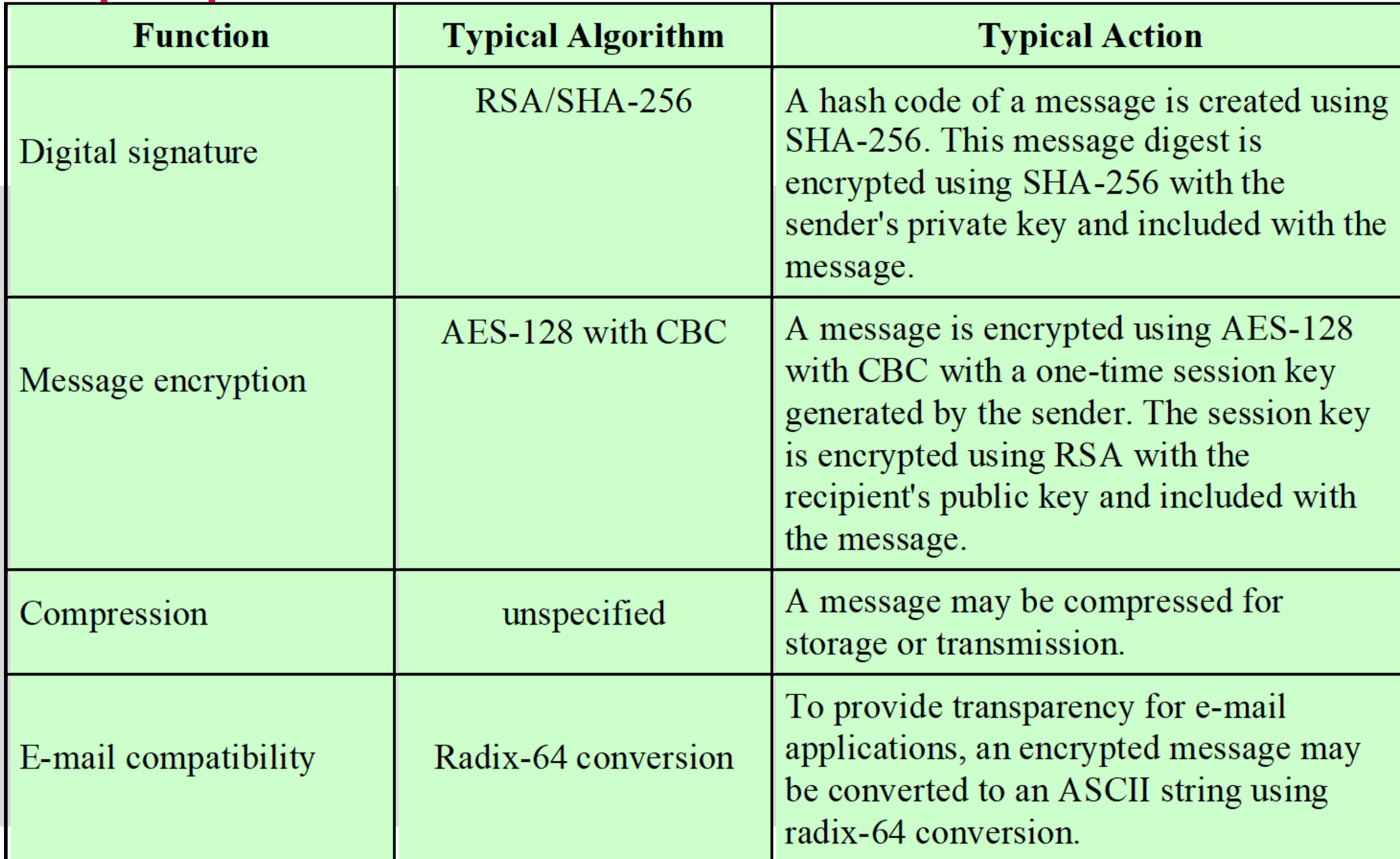
# 5. Secure/Multipurpose Internet Mail Extensions

## 1. S/MIME and PGP

- PGP
  - is based on «Web of Trust»: trusted third parties can certify keys, useful rather for closed groups.
- S/MIME
  - is based on hierarchical CA system similar to TLS.
- PGP and S/MIME
  - use different cryptographic schemes.
  - are not compatible.

# 5. Secure/Multipurpose Internet Mail Extensions

## 2. Services



| Function | Typical Algorithm | Typical Action |
|---|---|---|
| Digital signature | RSA/SHA-256 | A hash code of a message is created using SHA-256. This message digest is encrypted using SHA-256 with the sender's private key and included with the message. |
| Message encryption | AES-128 with CBC | A message is encrypted using AES-128 with CBC with a one-time session key generated by the sender. The session key is encrypted using RSA with the recipient's public key and included with the message. |
| Compression | unspecified | A message may be compressed for storage or transmission. |
| E-mail compatibility | Radix-64 conversion | To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII string using radix-64 conversion. |

# 5. Secure/Multipurpose Internet Mail Extensions

## 3. Authentication

by means of a digital signature

1. The sender creates a message.
2. SHA-256 is used to generate a 256-bit message digest of the message.
3. Message digest is encrypted with RSA using the sender's private key, result is appended to the message as well as the identifying information for the signer, which will enable the receiver to retrieve the signer's public key.
4. Receiver uses RSA with the sender's public key to decrypt and recover message digest.
5. Receiver generates a new message digest for the message and compares it with the decrypted hash code. If both match, the message is accepted as authentic.

# 5. Secure/Multipurpose Internet Mail Extensions

## 4. Confidentiality

S/MIME confidentiality by encrypting messages.

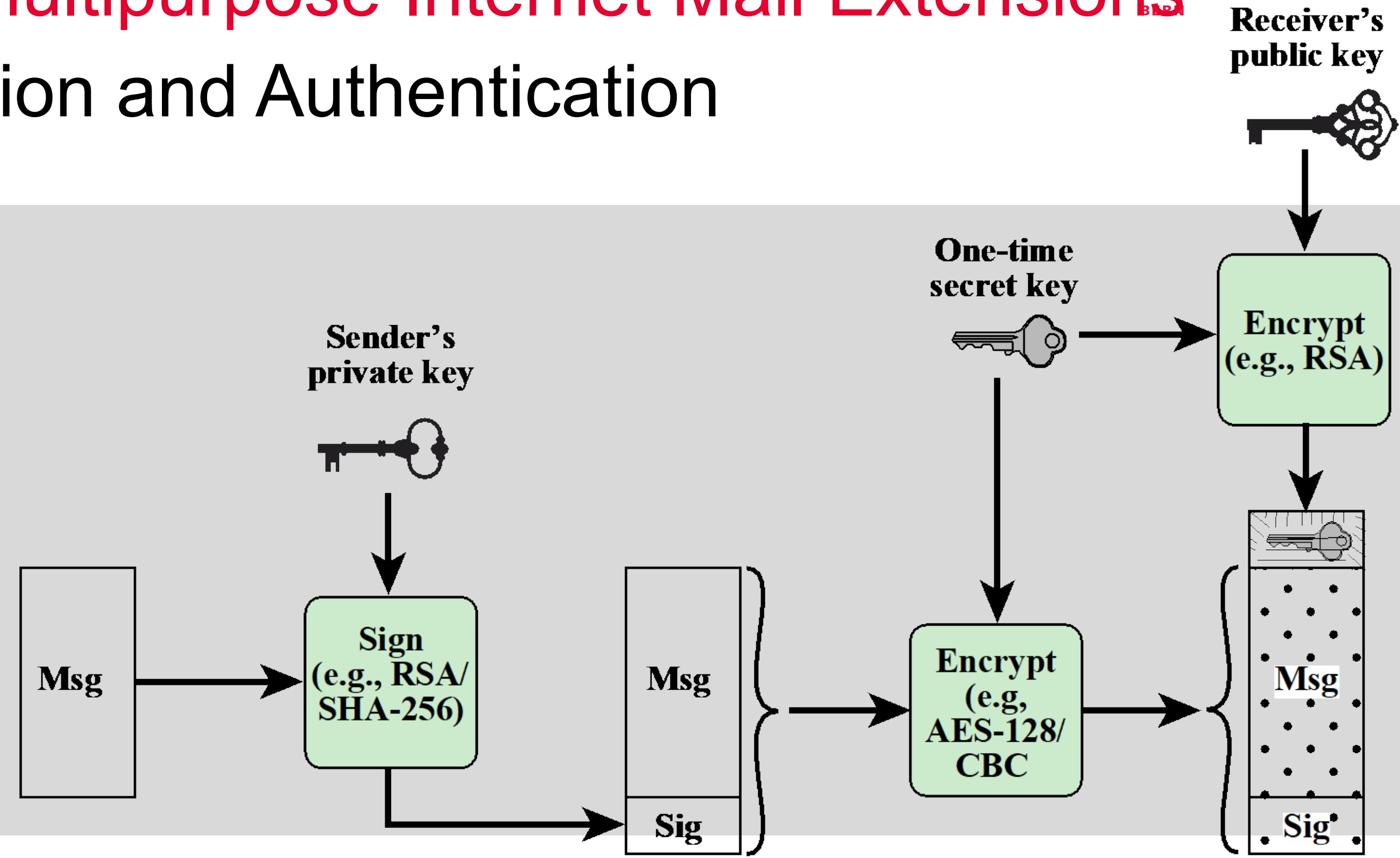
- Mostly: AES with a 128-bit key using CBC mode
- Key is also encrypted, typically with RSA
- Each symmetric key (content-encryption key) is used only once.

**Operation Sequence**

1. Sender generates a message and a random 128-bit number to be used as a content-encryption key for this message only.
2. Message is encrypted using the content-encryption key.
3. Content-encryption key is encrypted with RSA using the recipient's public key and is attached to the message.
4. Receiver uses RSA with its private key to decrypt and recover content-encryption key.
5. Content-encryption key is used to decrypt message.

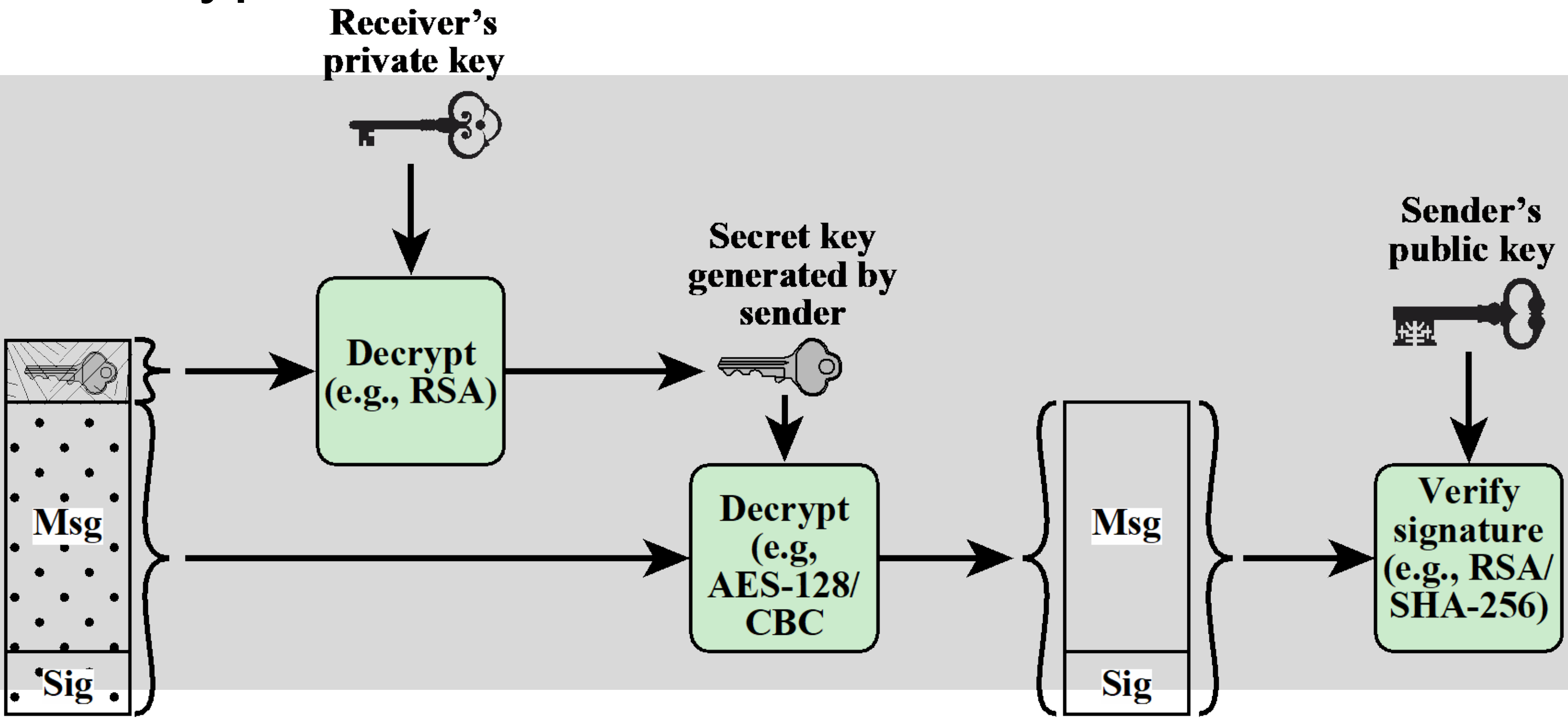
# 5. Secure/Multipurpose Internet Mail Extensions

## 4.1 Encryption and Authentication



# 5. Secure/Multipurpose Internet Mail Extensions

## 4.2 Decryption and Authentication


Receiver's private key
Decrypt (e.g., RSA)
Secret key generated by sender
Sender's public key
Msg
Sig
Decrypt (e.g, AES-128/ CBC
Msg
Sig
Verify signature (e.g., RSA/ SHA-256)

# 5. Secure/Multipurpose Internet Mail Extensions

## 5.1 S/MIME Message Content Types (RFC 5652)

- Data
  - refers to inner MIME-encoded message content, which may then be encapsulated in a SignedData, EnvelopedData, or CompressedData content type
- EnvelopedData
  - consists of encrypted content of any type and encryption keys for one or more recipients
- SignedData
  - used to apply a digital signature to a message
- CompressedData
  - used to apply data compression to a message
- Clear signing
  - A digital signature is calculated for a MIME-encoded message and the two parts (message and signature) form a multipart MIME message
  - can be read and their signatures verified by email entities that do not implement S/MIME
- For most cases: result of security algorithm will be arbitrary binary data → base64

# 5. Secure/Multipurpose Internet Mail Extensions

## 5.2 Enveloped Data

1. Generate a pseudorandom session key for a particular symmetric encryption algorithm
2. For each recipient, encrypt the session key with the recipient's public RSA key.
3. For each recipient, prepare a block known as RecipientInfo containing
   - identifier of recipient's public-key certificate
   - identifier of the algorithm used to encrypt the session key
   - encrypted session key
4. Encrypt message content with session key.

```
Content-Type: application/pkcs7-mime;
smime-type=enveloped-data; name=smime.p7m

Content-Transfer-Encoding: base64

Content-Disposition: attachment;
filename=smime.p7m
```

```
rfvbnj756tbBghyHhHUujhJhjH77n8HHGT9HG4VQpfyF
467GhIGfHfYT6
7n8HHGghyHhHUujhJh4VQpfyF467GhIGfHfYGTrfvbnj
T6jH7756tbB9H
f8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6g
hyHhHUujpfyF4
0GhIGfHfQbnj756YT64V
```

# 5. Secure/Multipurpose Internet Mail Extensions

## 5.3 Signed Data

1. Select message digest algorithm (SHA or MD5).
2. Compute message digest of content to be signed.
3. Encrypt message digest with signer’s private key.
4. Prepare a block known as SignerInfo containing
   - signer’s public-key certificate
   - identifier of message digest algorithm
   - identifier of algorithm used to encrypt the message digest
   - encrypted message digest.

```
Content-Type: application/pkcs7-mime;
smime-type=signed-data; name=smime.p7m

Content-Transfer-Encoding: base64

Content-Disposition: attachment;
filename=smime.p7m
```

```
567GhIGfHfYT6ghyHhHUujpfyF4f8HHGTrfvhJhjH776tbB9
HG4VQbnj7
77n8HHGT9HG4VQpfyF467GhIGfHfYT6rfvbnj756tbBghyHh
HUujhJhjH
HUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H7n
8HHGghyHh
6YT64V0GhIGfHfQbnj75
```

# 5. Secure/Multipurpose Internet Mail Extensions

## 5.4 Clear Signing

- achieved using multipart content type with signed subtype
- Signing process does not involve transforming the message to be signed.
- Recipients with MIME but not S/MIME capability can read incoming message.

```
Content-Type: multipart/signed;
protocol=”application/pkcs7-signature”; micalg=sha1;
boundary=boundary42

—boundary42

Content-Type: text/plain This is a clear-signed message.

—boundary42

Content-Type: application/pkcs7-signature; name=smime.p7s

Content-Transfer-Encoding: base64

Content-Disposition: attachment; filename=smime.p7s

ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj
n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
7GhIGfHfYT64VQbnj756

—boundary42—
```

# 5. Secure/Multipurpose Internet Mail Extensions

## 6.1 Certificate Processing

- S/MIME uses public-key certificates that conform to version 3 of X.509
- S/MIME managers and/or users must configure each client with a list of trusted keys and certificate revocation lists
  - The responsibility is local for maintaining the certificates needed to verify incoming signatures and to encrypt outgoing messages.
- Certificates are signed by certification authorities.

# 5. Secure/Multipurpose Internet Mail Extensions

## 6.2 User Agent Role

- Key generation
  - The user of some related administrative utility
    - MUST be capable of generating separate DH and DSS key pairs.
    - SHOULD be capable of generating RSA key pairs.
- Registration
  - A user’s public key must be registered with CA in order to receive an X.509 public-key certificate.
- Certificate storage and retrieval
  - A user requires access to a local list of certificates (to be maintained by user of administrative entity) in order to verify incoming signatures and to encrypt outgoing messages.

# 6. DNS Security

## 1.1 Domain Name System

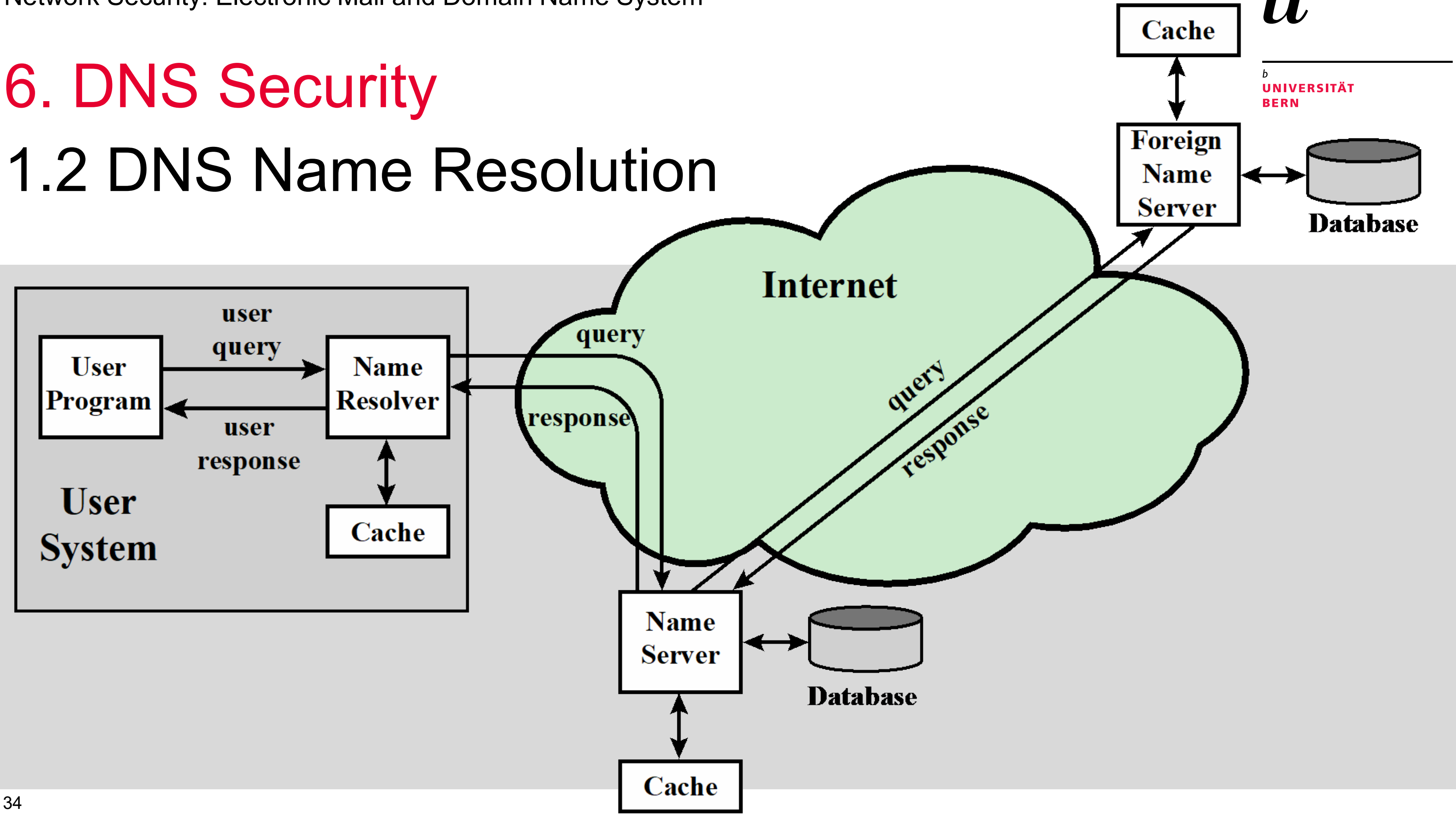
- Directory lookup service mapping the name of a host to its numeric IP address
- is used by MUAs and MTAs to find the address of the next hop server for mail delivery
- is based on a hierarchical database containing Resource Records that include name, IP address, and other information about hosts

Key features of DNS database are:

- variable-depth hierarchy for names
- distributed database

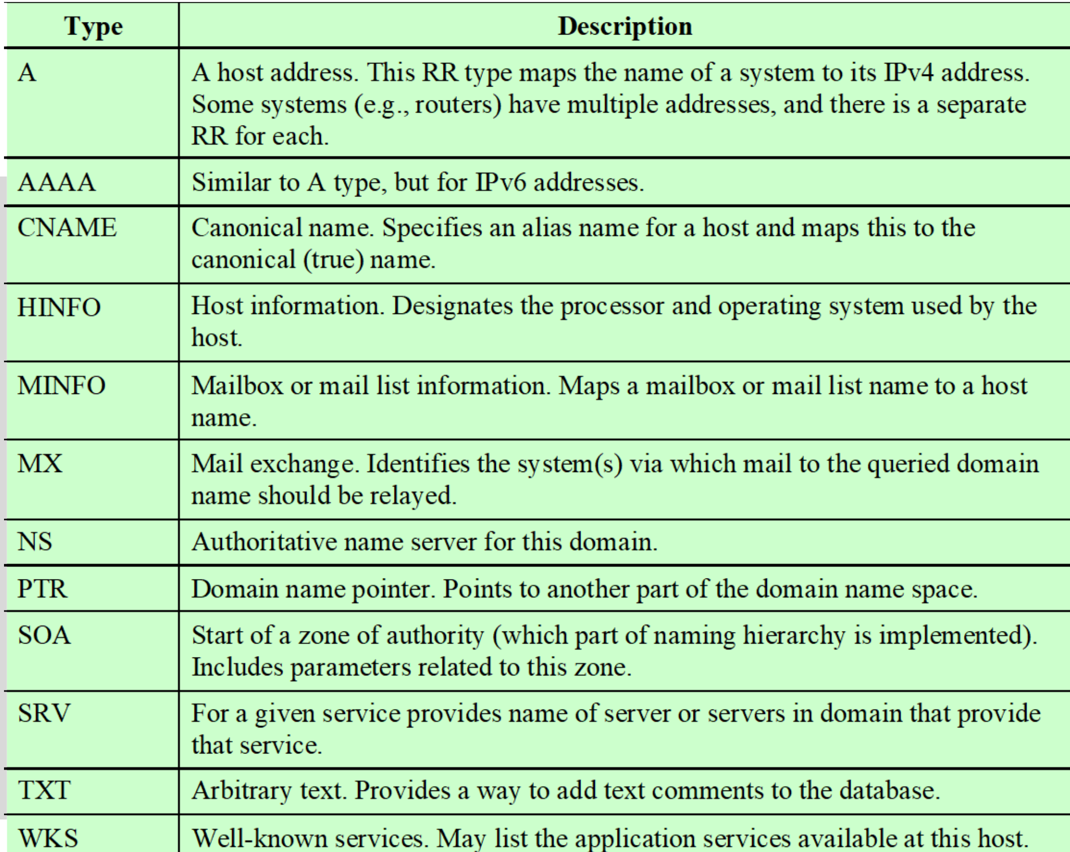
# 6. DNS Security

## 1.2 DNS Name Resolution



# 6. DNS Security

## 1.3 DNS Resource Record Types



| Type | Description |
|---|---|
| A | A host address. This RR type maps the name of a system to its IPv4 address. Some systems (e.g., routers) have multiple addresses, and there is a separate RR for each. |
| AAAA | Similar to A type, but for IPv6 addresses. |
| CNAME | Canonical name. Specifies an alias name for a host and maps this to the canonical (true) name. |
| HINFO | Host information. Designates the processor and operating system used by the host. |
| MINFO | Mailbox or mail list information. Maps a mailbox or mail list name to a host name. |
| MX | Mail exchange. Identifies the system(s) via which mail to the queried domain name should be relayed. |
| NS | Authoritative name server for this domain. |
| PTR | Domain name pointer. Points to another part of the domain name space. |
| SOA | Start of a zone of authority (which part of naming hierarchy is implemented). Includes parameters related to this zone. |
| SRV | For a given service provides name of server or servers in domain that provide that service. |
| TXT | Arbitrary text. Provides a way to add text comments to the database. |
| WKS | Well-known services. May list the application services available at this host. |

# 6. DNS Security

## 1.4 DNS Resource Records

Server stores Resource Records (Name, Value, Type, Class, Lifetime)

Examples

(inf.unibe.ch, asterix.unibe.ch, NS, IN)

(inf.unibe.ch, obelix.unibe.ch, MX, IN)

(asterix.unibe.ch, 130.92.64.4, A, IN)

(obelix.unibe.ch, 130.92.64.5, A, IN)

(_http._tcp.inf.unibe.ch, 80 www.inf.unibe.ch, SRV, IN)

# 6. DNS Security

## 2.1 DNS Security Extensions

- provide end-to-end protection by digital signatures that are created by responding zone administrators and verified by a recipient’s resolver software.
- avoid need to trust intermediate name servers and resolvers that cache or route the DNS records originating from the responding zone administrator before they reach the source of the query.
- consist of a set of new resource record types and modifications to the existing DNS protocol.
- are defined in RFCs 4033-4035.

# 6. DNS Security

## 2.2 DNSSEC RRs

1. DNSKEY contains a public key
   - Asymmetric key pair exists for each zone. Private key is used for signatures.
2. Resource Record Digital Signature
   - associated with each Resource Record Set (RRs with same label, class, type)
   - When client requests data, RRset is returned with associated digital signature in RRSIG
3. NSEC: authenticated denial of existence record
   - To secure all DNS lookups, DNSSEC uses the NSEC RR to authenticate negative responses to queries.
   - NSEC is used to identify the range of DNS names or resource record types that do not exist among the sequence of domain names in a zone.
4. Delegation Signer
   - Hash of public key, stored on higher level
   - facilitates key signing and authentication between DNS zones to create an authentication chain from the root of the DNS tree down to a specific domain name

# 6. DNS Security

## 2.2.1 Operation

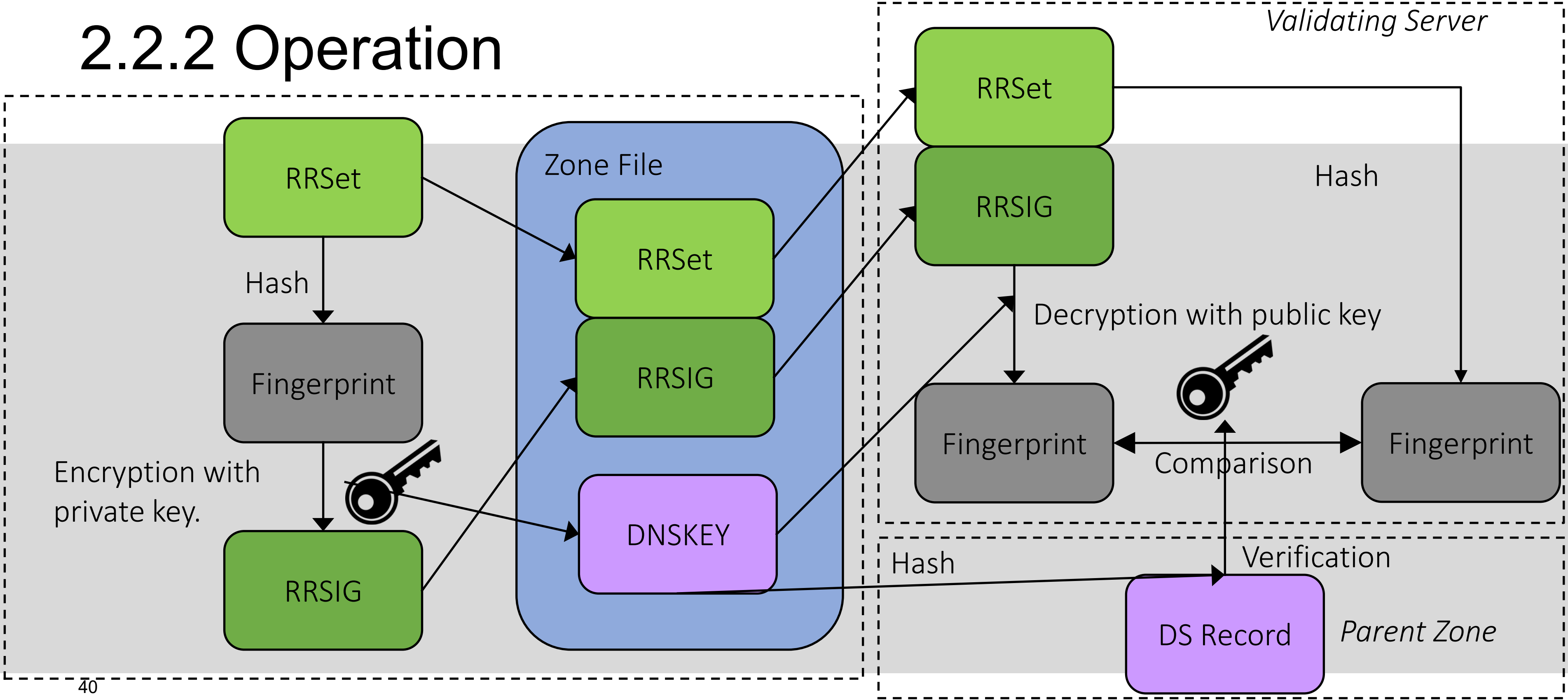
- DNSSEC Operation
  - Data origin authentication ensures that data originated from correct source.
  - Data integrity verification ensures that content of a RR has not been modified.
- DNS zone administrator
  - digitally signs every Resource Record set in the zone, e.g.,
    - www.example.org IN A 127.0.0.1
    - www.example.org IN A 192.168.0.1
  - publishes this collection of digital signatures, along with the zone administrator’s public key, in the DNS itself.

Trust in the source’s public key (for signature verification) is established

- not by going to a third party or a chain of third parties (as in PKI chaining),
- but by starting from a trusted zone (such as the root zone) and establishing the chain of trust down to the current source of response through successive verifications of signature of the public key of a child by its parent.

# 6. DNS Security

## 2.2.2 Operation



# 7. DNS-based Authentication of Named Entities

## 1. Overview

DANE

- addresses the vulnerability of the use of CAs in a global PKI.
- allows X.509 certificates, commonly used for TLS, to be bound to DNS names using DNSSEC.
- is a way to authenticate TLS client and server entities without CA.
- has been defined in RFC 6698.

# 7. DNS-based Authentication of Named Entities

## 2. TLS Authentication RR

- can be used to securely authenticate TLS certificates.
- specifies that a service certificate or a CA can be authenticated in the DNS itself.
- enables certificate issue and delivery to be tied to a given domain.
- A server domain owner creates a TLSA RR that identifies the certificate and its public key.
- When a client receives an X.509 certificate in the TLS negotiation, it looks up the TLSA RR for that domain and matches the TLSA data against the certificate as part of the client’s certificate validation procedure.

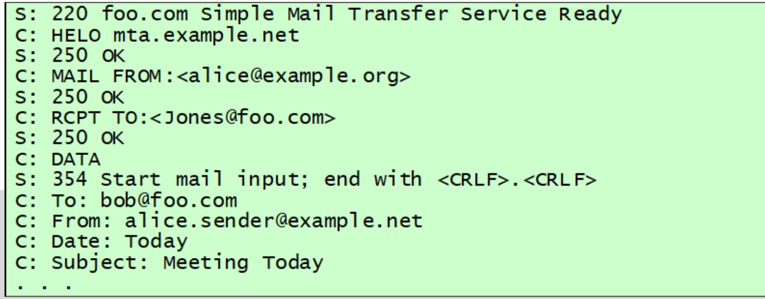
# 7. DNS-based Authentication of Named Entities

## 3. DANE for SMTP

- can be used in conjunction with SMTP over TLS, as provided by STARTTLS.
- can authenticate the certificate of the SMTP server that the user’s MUA communicates with.
- can also authenticate the TLS connections between SMTP MTAs.

# 8. Sender Policy Framework

## 1. Overview



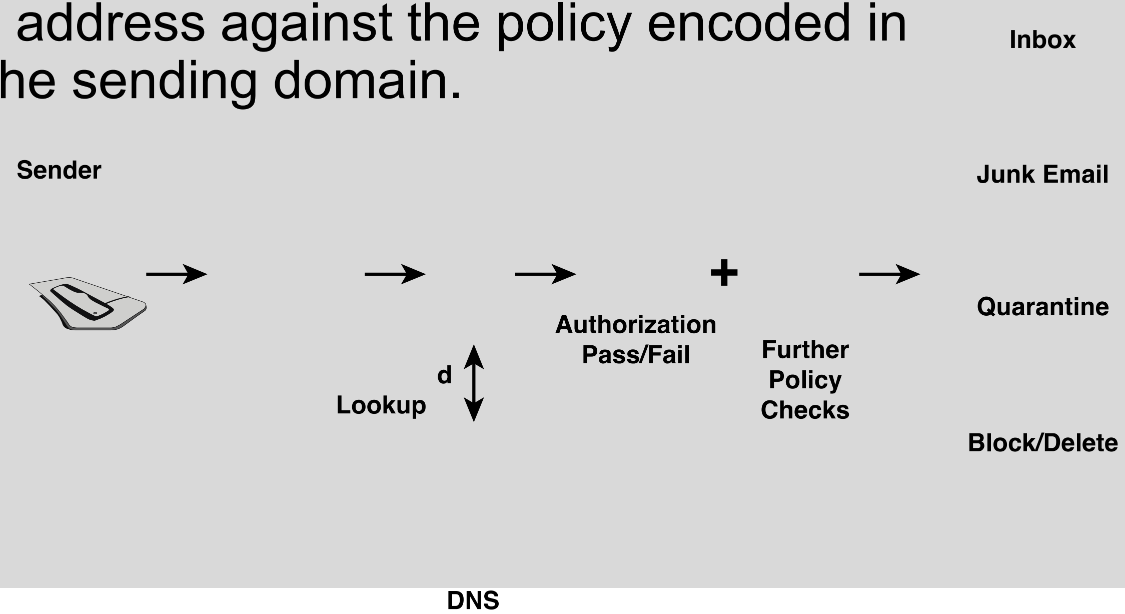
```
S: 220 foo.com Simple Mail Transfer Service Ready
C: HELO mta.example.net
S: 250 OK
C: MAIL FROM:<alice@example.org>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: To: bob@foo.com
C: From: alice.sender@example.net
C: Date: Today
C: Subject: Meeting Today
. . .
```

- SPF is way for a sending domain to identify and assert the mail senders for a given domain.
- Problem: With the current email infrastructure, any host can use any domain name for each of the various identifiers in the mail header, not just the domain name where the host is located. Drawbacks of this approach:
  - It is difficult for mail handlers to filter out emails.
  - Entities can make use of email providers’ domain names, often with malicious intent.
- RFC 7208 provides a protocol by which email providers can authorize hosts to use their domain names in the “MAIL FROM” or “HELO” identities.
- Email providers publish SPF records in the DNS specifying which hosts are permitted to use their names
- Mail receivers use published SPF records to test authorization of sending MTAs using a given “HELO” or “MAIL FROM” identity during a mail transaction.

# 8. Sender Policy Framework

## 2. Operation

SPF checks a sender’s IP address against the policy encoded in any SPF record found at the sending domain.



# 8. Sender Policy Framework

## 3. Sender Side

- A sending domain needs to identify all the senders for a given domain and add that information into the DNS as a separate resource record.
- Sending domain encodes the appropriate policy for each sender using the SPF syntax by using TXT RRs.
- Mechanisms are used to define an IP address or range of addresses to be matched.
- Modifiers indicate the policy for a given match.

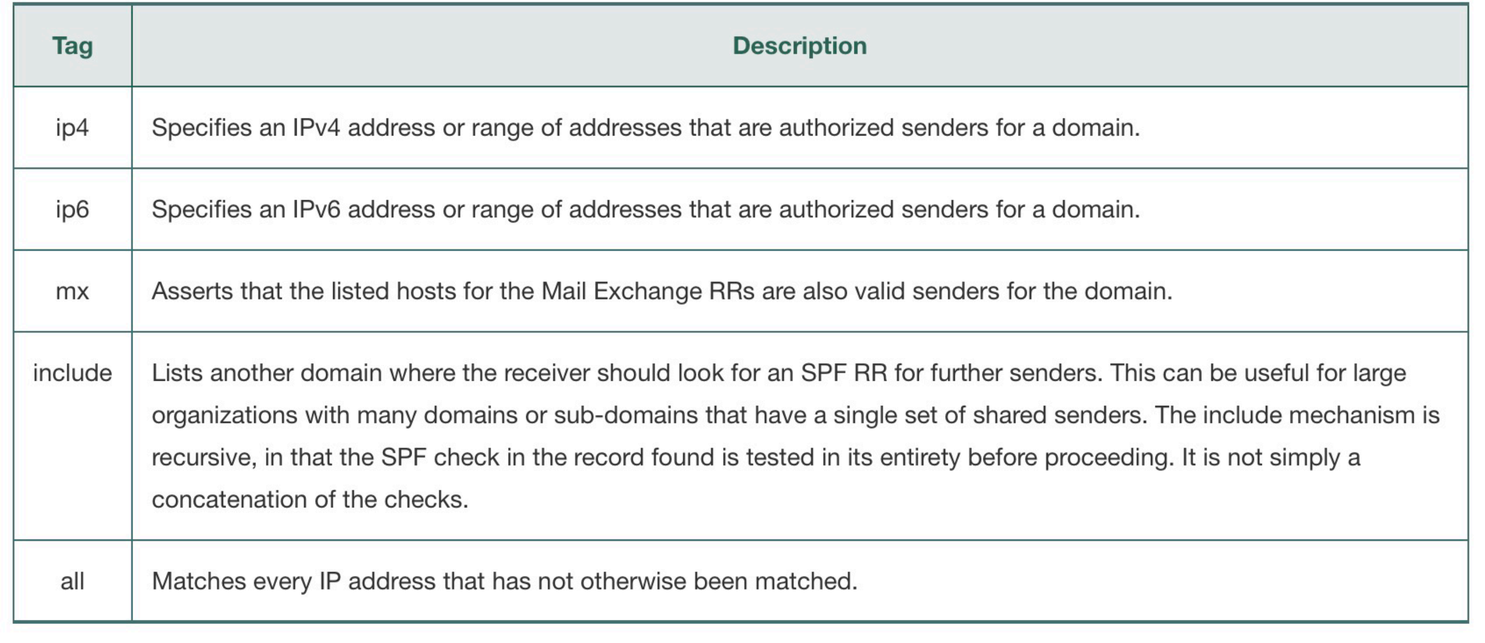
# 8. Sender Policy Framework

## 4. Receiver Side

- If SPF is implemented at a receiver, the SPF entity uses the SMTP envelope **`MAIL FROM: address domain`** and the IP address of the sender to query an SPF TXT RR.
- SPF checks can be started before the body of the email message is received.
- Alternatively, the entire message can be absorbed and buffered until all the checks are finished.
- In either case, checks must be completed before the mail message is sent to the end user’s inbox.

# 8. Sender Policy Framework

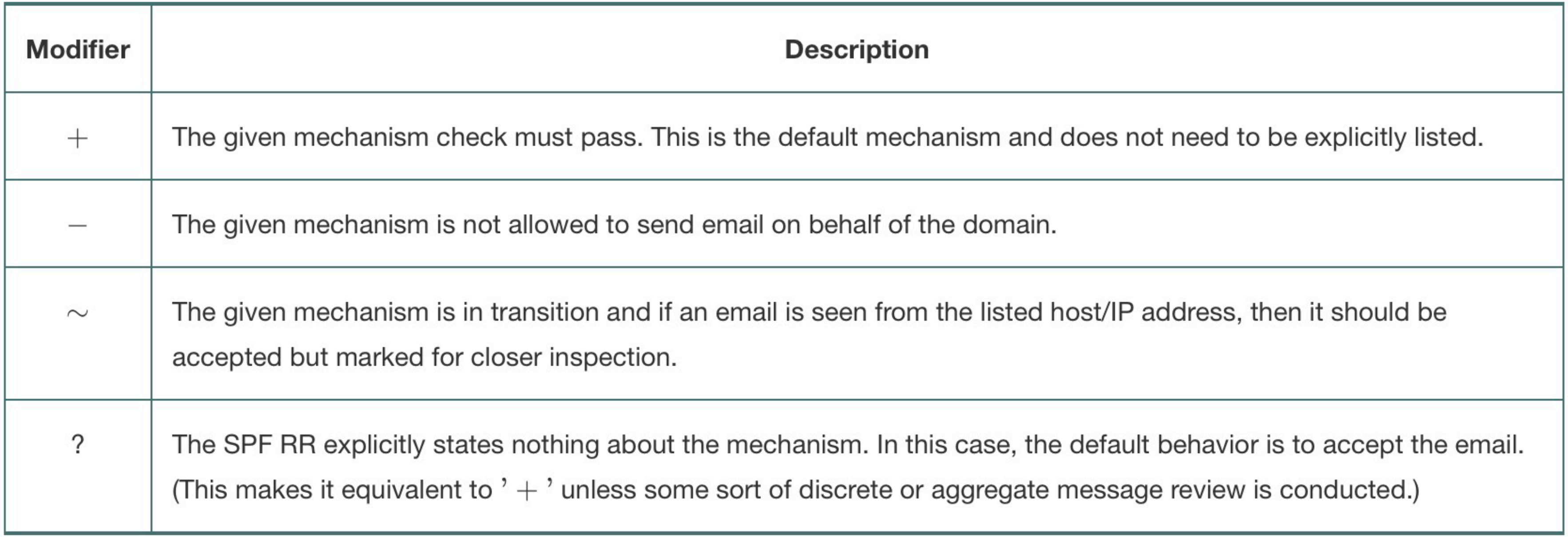
## 5. Mechanisms



| Tag | Description |
|---|---|
| ip4 | Specifies an IPv4 address or range of addresses that are authorized senders for a domain. |
| ip6 | Specifies an IPv6 address or range of addresses that are authorized senders for a domain. |
| mx | Asserts that the listed hosts for the Mail Exchange RRs are also valid senders for the domain. |
| include | Lists another domain where the receiver should look for an SPF RR for further senders. This can be useful for large organizations with many domains or sub-domains that have a single set of shared senders. The include mechanism is recursive, in that the SPF check in the record found is tested in its entirety before proceeding. It is not simply a concatenation of the checks. |
| all | Matches every IP address that has not otherwise been matched. |

# 8. Sender Policy Framework

## 6. Modifiers

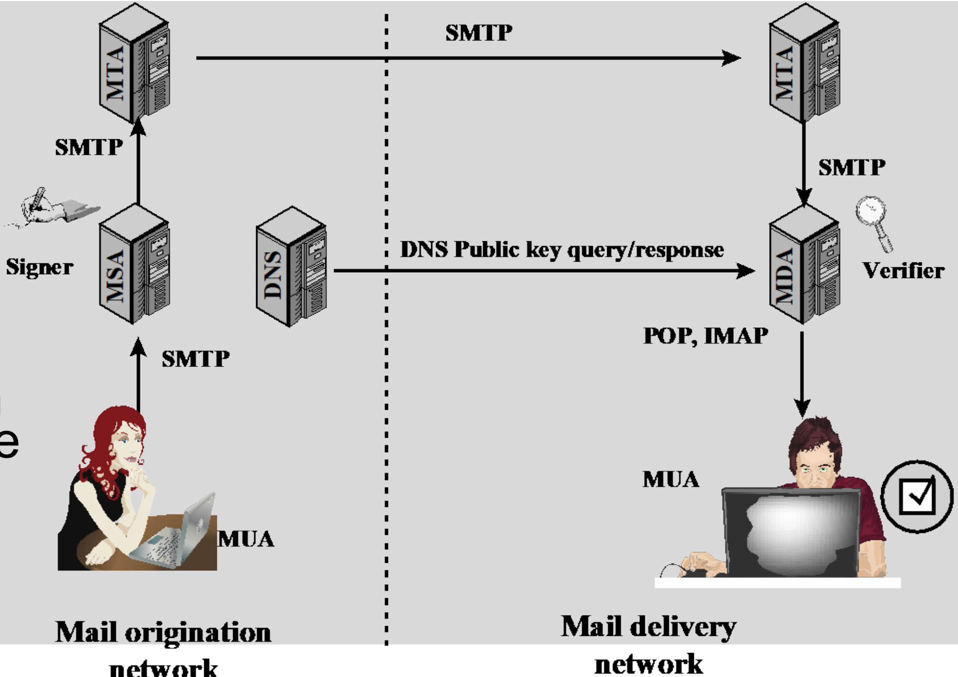


| Modifier | Description |
|---|---|
| + | The given mechanism check must pass. This is the default mechanism and does not need to be explicitly listed. |
| – | The given mechanism is not allowed to send email on behalf of the domain. |
| ~ | The given mechanism is in transition and if an email is seen from the listed host/IP address, then it should be accepted but marked for closer inspection. |
| ? | The SPF RR explicitly states nothing about the mechanism. In this case, the default behavior is to accept the email. (This makes it equivalent to ’ + ’ unless some sort of discrete or aggregate message review is conducted.) |

# 9. Domain Keys Identified Mail

## 1. Overview

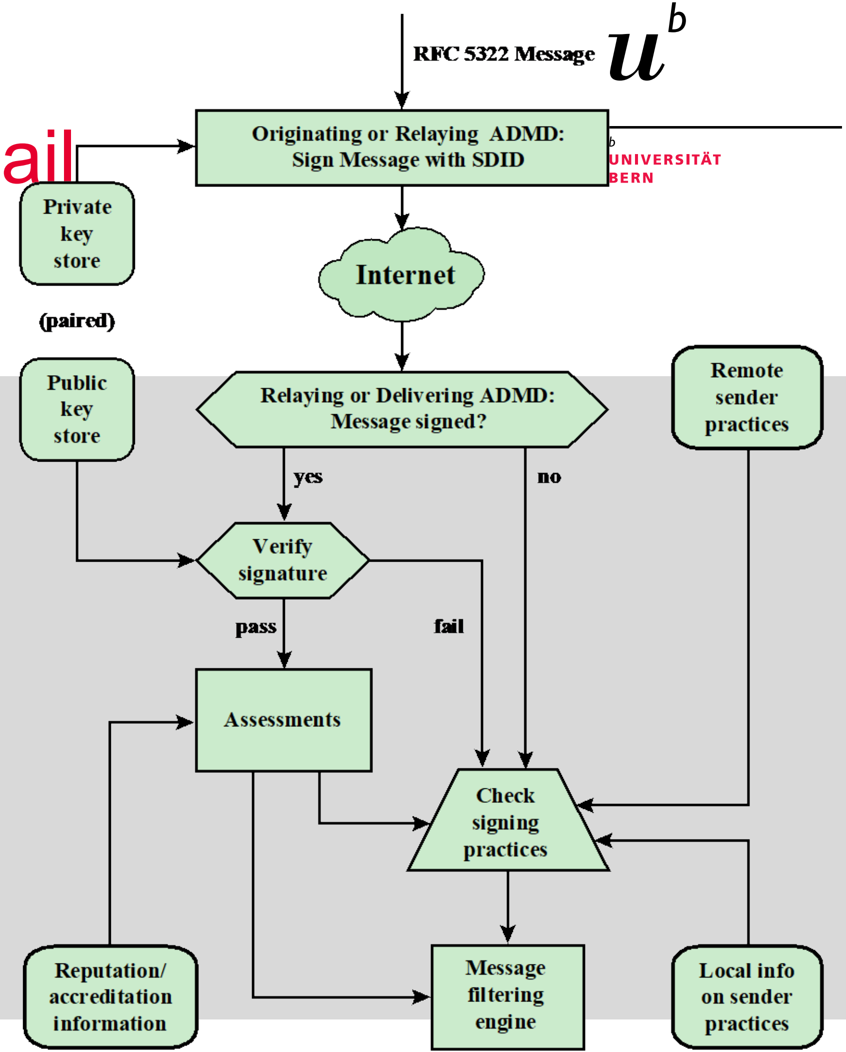
- A specification for cryptographically signing e-mail messages, permitting a signing domain to claim responsibility for a message.
- Message recipients can verify the signature by querying the signer's domain to retrieve the appropriate public key and can confirm that the message was attested by a party in possession of the private key for the signing domain.
- Target: spam email prevention
- RFC 6376



# 9. Domain Keys Identified Mail

## 2. Functional Flow

- Signing might be performed by MUA, MSA, or MTA.
- Verifying might be performed by MTA, MDA, or MUA.
- If the signature passes, reputation information is used to assess the signer and that information is passed to the message filtering system.
- If signature fails or there is no signature using the author's domain, information about signing practices related to the author can be retrieved remotely and/or locally, and that information is passed to the message filtering system. For example, if the sender (e.g., gmail) uses DKIM but no DKIM signature is present, then the message may be considered fraudulent.
- Signature is inserted into the RFC 5322 message as an additional header entry, starting with the keyword DKIM-Signature.



# 10. Domain-based Message Authentication, Reporting, and Conformance

## 1. Overview

**Problem**

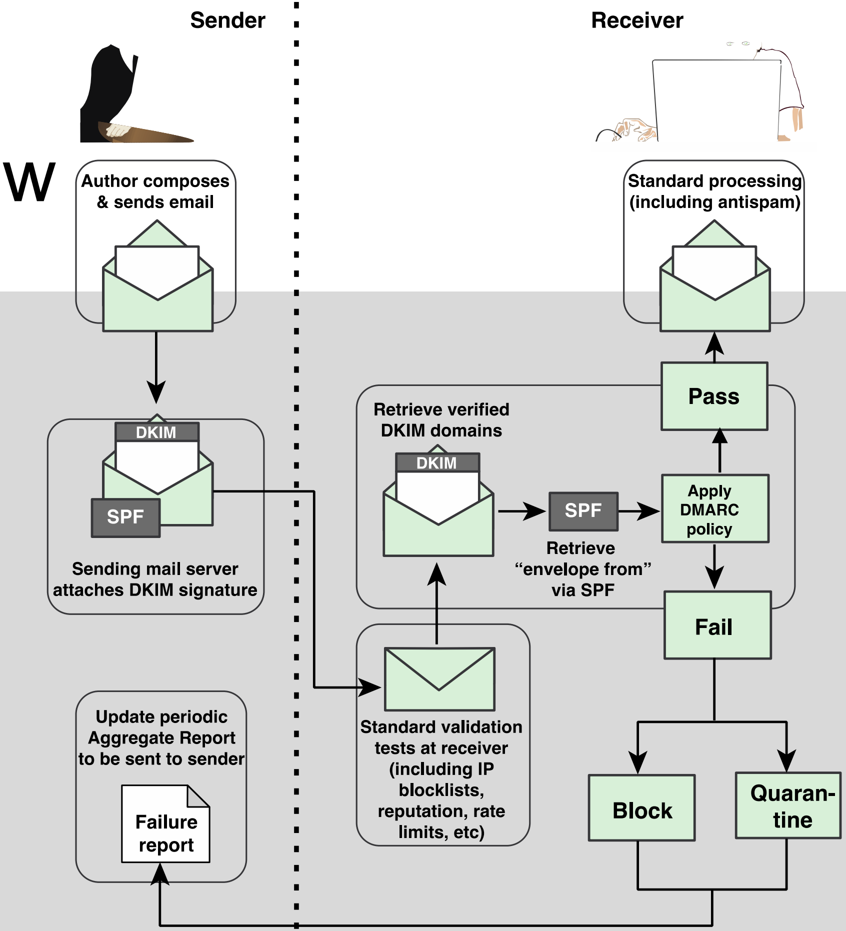
- Neither SPF nor DKIM include a mechanism to tell receivers if SPF or DKIM are in use.

**DMARC** (RFC 7489)

- allows email senders to specify policy on how their mail should be handled, the types of reports that receivers can send back, and the frequency those reports should be sent.
- policies in DNS TXT records
- works with DKIM and SPF.

# 10. DMARC

## 2. Functional Flow


Sender
Receiver
Author composes & sends email
DKIM
SPF
Sending mail server attaches DKIM signature
Update periodic Aggregate Report to be sent to sender
Failure report
Standard validation tests at receiver (including IP blocklists, reputation, rate limits, etc)
Retrieve verified DKIM domains
DKIM
SPF
Retrieve "envelope from" via SPF
Apply DMARC policy
Pass
Fail
Block
Quaran-tine
Standard processing (including antispam)

# Thanks a lot

# for your Attentation

**Prof. Dr. Torsten Braun, Institut für Informatik**

Bern, 09.05.2022 – 16.05.2022