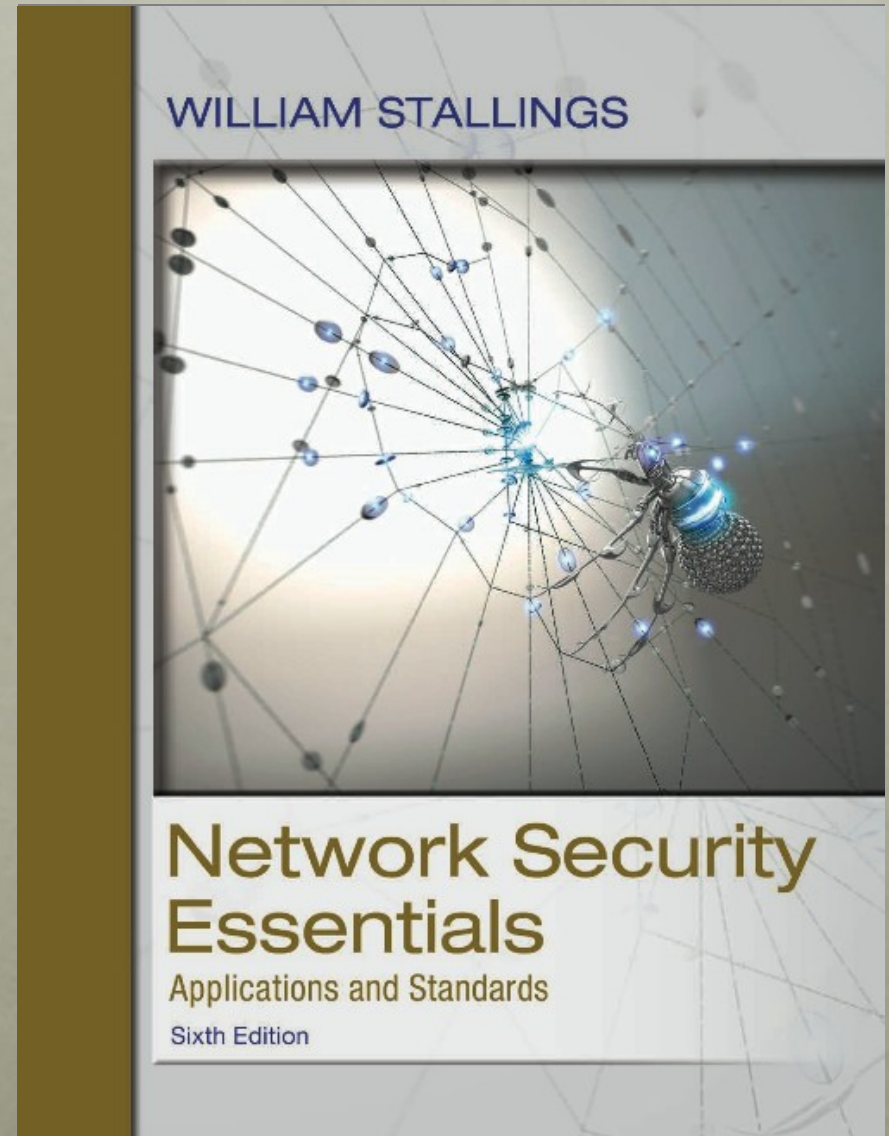


# Network Security Essentials

Sixth Edition

by William Stallings

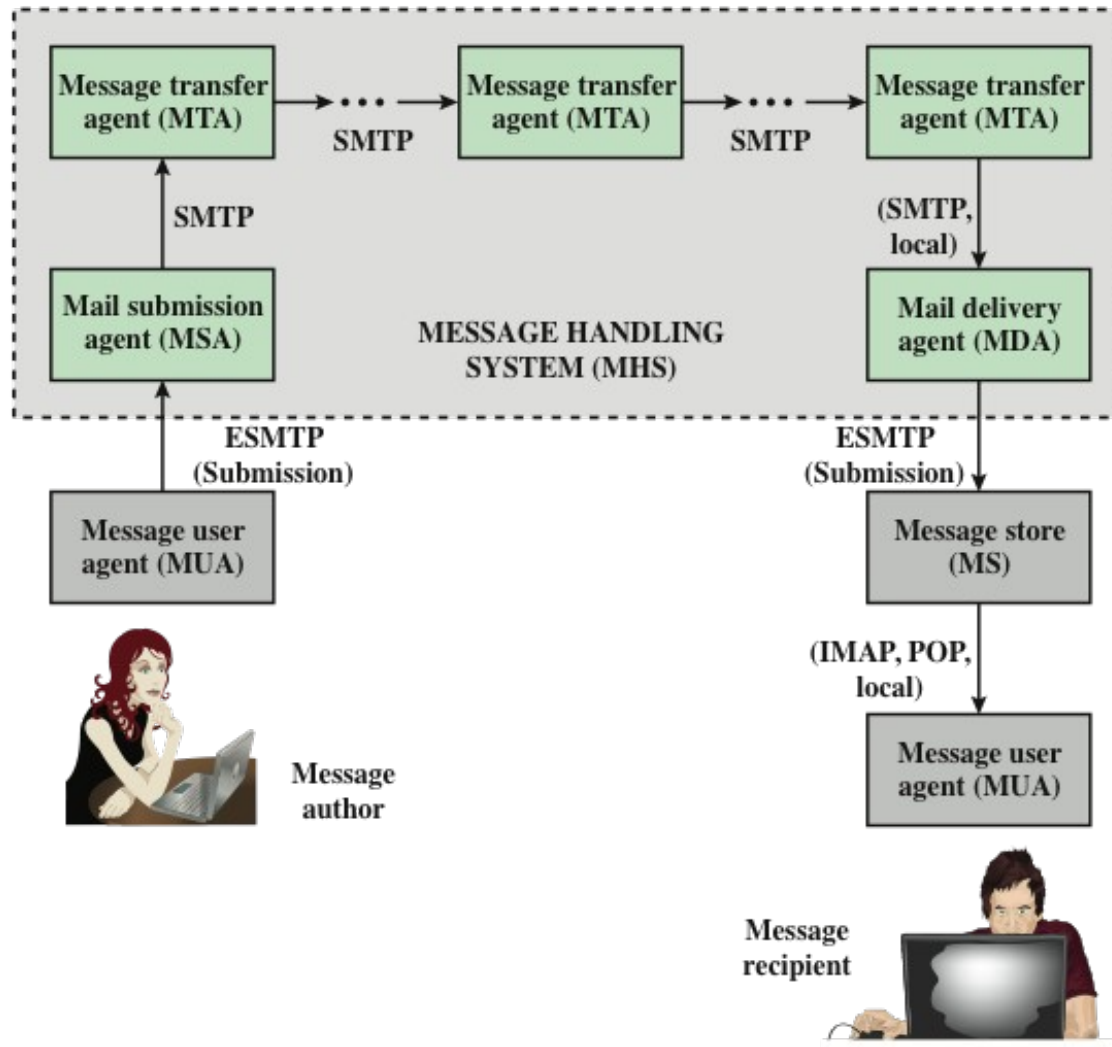


# Chapter 8

## Electronic Mail Security

# Internet mail architecture

- Defined in RFC 5598 (*Internet Mail Architecture*, July 2009)
- E-mail components:
  - At its most fundamental level, the internet mail architecture consists of a user world in the form of Message User Agents (MUA)
  - And the transfer world, in the form of the Message Handling Service (MHS), which is composed of Message Transfer Agents (MTA)
    - The MHS accepts a message from one user and delivers it to one or more other users, creating a virtual MUA-to-MUA exchange environment
    - This architecture involves three types of interoperability



**Figure 8.1 Function Modules and Standardized Protocols  
Used Between Them in the Internet Mail Architecture**



# E-Mail components

- Administrative management domain (ADMD)
  - Internet e-mail provider
  - Examples include a department that operates a local mail relay, an IT department that operates an enterprise mail relay, and an ISP that operates a public shared e-mail service
  - Each ADMD can have different operating policies and trust-based decision making
- Domain name system (DNS)
  - A directory lookup service that provides a mapping between the name of a host on the Internet and its numerical address



# E-Mail Protocols

## SMTP

- Simple Mail Transfer Protocol (SMTP)
  - Encapsulates an e-mail message in an envelope and is used to relay the encapsulated messages from source to destination through multiple MTAs
  - Was originally specified in 1982 as RFC 821
    - Has undergone several revisions, the most current being RFC5321 (October 2008)
  - Is a text-based client-server protocol where the client (e-mail sender) contacts the server (next-hop recipient) and issues a set of commands to tell the server about the message to be sent, then sending the message itself
    - The majority of these commands are ASCII text messages sent by the client and a resulting return code returned by the server

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: HELO bar.com
S: 250 OK
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT TO:<Green@foo.com>
S: 550 No such user here
C: RCPT TO:<Brown@foo.com>
S: 250 OK
C: DATA
S: 354 start mail input; end with <CRLF>.<CRLF>
C: Blah blah blah...
C: ...etc. etc. etc.
C: <CRLF><CRLF>
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

**Figure 8.2 Example SMTP Transaction Scenario**

# Mail access protocols

## Post Office Protocol (POP3)

- Allows an e-mail client (user agent) to download an e-mail from an e-mail server (MTA)
- POP3 user agents connect via TCP to the server (usually port 110)
- The user agent enters a username and password
- After authorization, the UA can issue POP3 commands to retrieve and delete mail

## Internet Mail Access Protocol (IMAP)

- Enables an e-mail client to access mail on an e-mail server
- Uses TCP, with server TCP port 143
- Is more complex than POP3
- Provides stronger authentication than POP3 and provides other functions not supported by POP3



# E-mail formats

- RFC 5322
  - Defines a format for text messages that are sent using electronic mail
  - Has been the standard for Internet-based text mail messages
  - Messages are viewed as having an envelope and contents
    - The envelope contains whatever information is needed to accomplish transmission and delivery
    - The contents compose the object to be delivered to the recipient
    - RFC 5322 standard applies only to the contents

# E-mail formats

- Multipurpose Internet Mail Extensions (MIME)
  - Extension to the RFC 5322 framework
  - RFCs 2045 through 2049 define MIME
  - The MIME specification includes the following elements:
    - Five new message header fields are defined, which may be included in an RFC 5322 header, providing information about the body of the message
    - A number of content formats are defined, thus standardizing representations that support multimedia electronic mail
    - Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system

# Header fields defined in mime

- MIME-Version
  - This field indicates that the message conforms to RFCs 2045 and 2046
- Content-Type
  - Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to represent the data to the user or otherwise deal with the data in an appropriate manner
- 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
- Content-ID
  - Used to identify MIME entities uniquely in multiple contexts
- Content-Description
  - A text description of the object with the body

**Table 8.1**

# **MIME Content Types**

(Table can be found on page 245 in the textbook)

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.

# Table 8.2

## MIME Transfer Encodings

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.

(Table can be found on page 248 in the textbook)



```

MIME-Version: 1.0
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: A multipart example
Content-Type: multipart/mixed;
    boundary=unique-boundary-1
This is the preamble area of a multipart message. Mail readers that understand multipart format should ignore this preamble.
If you are reading this text, you might want to consider changing to a mail reader that understands how to properly display
multipart messages.

--unique-boundary-1
...Some text appears here...
[Note that the preceding blank line means no header fields were given and this is text, with charset US ASCII. It could have
been done with explicit typing as in the next part.]

--unique-boundary-1
Content-type: text/plain; charset=US-ASCII
This could have been part of the previous part, but illustrates explicit versus implicit typing of body parts.

--unique-boundary-1
Content-Type: multipart/parallel;    boundary=unique-boundary-2

--unique-boundary-2
Content-Type: audio/basic
Content-Transfer-Encoding: base64
... base64-encoded 8000 Hz single-channel mu-law-format audio data goes here....

--unique-boundary-2
Content-Type: image/jpeg
Content-Transfer-Encoding: base64
... base64-encoded image data goes here....

--unique-boundary-2--
--unique-boundary-1
Content-type: text/enriched

This is <bold><italic>richtext.</italic></bold> <smaller>as defined in RFC 1896</smaller>

Isn't it <bigger><bigger>cool?</bigger></bigger>

--unique-boundary-1
Content-Type: message/rfc822

From: (mailbox in US-ASCII)
To: (address in US-ASCII)
Subject: (subject in US-ASCII)
Content-Type: Text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: Quoted-printable

... Additional text in ISO-8859-1 goes here ...

--unique-boundary-1--

```

**Figure 8.3 Example MIME Message Structure**

(Figure 8.3 can be found on page 249 in the textbook)

# Canonical and Native Forms

Canonical form is a format, appropriate to the content type, that is standardized for use between systems

- The entire body, including out-of-band information such as record lengths and possibly file attribute information, is converted to a universal canonical form. The specific media type of the body as well as its associated attributes dictates the nature of the canonical form that is used. Conversion to the proper canonical form may involve character set conversion, transformation of audio data, compression, or various other operations specific to the various media types.

Native form is a format that may be peculiar to a particular system

- The body to be transmitted is created in the system's native format. The native character set is used and, where appropriate, local end-of-line conventions are used as well. The body may be any format that corresponds to the local model for the representation of some form of information.

# E-mail threats

E-mail security threats can be classified as:

## Authenticity-related threats

- Could result in unauthorized access to an enterprise's e-mail system

## Integrity-related threats

- Could result in unauthorized modification of e-mail content

## Confidentiality-related threats

- Could result in unauthorized disclosure of sensitive information

## Availability-related threats

- Could prevent end users from being able to send or receive e-mail

# Table 8.3

## Email Threats and Mitigations

(Table can be found on page 251 in the textbook)

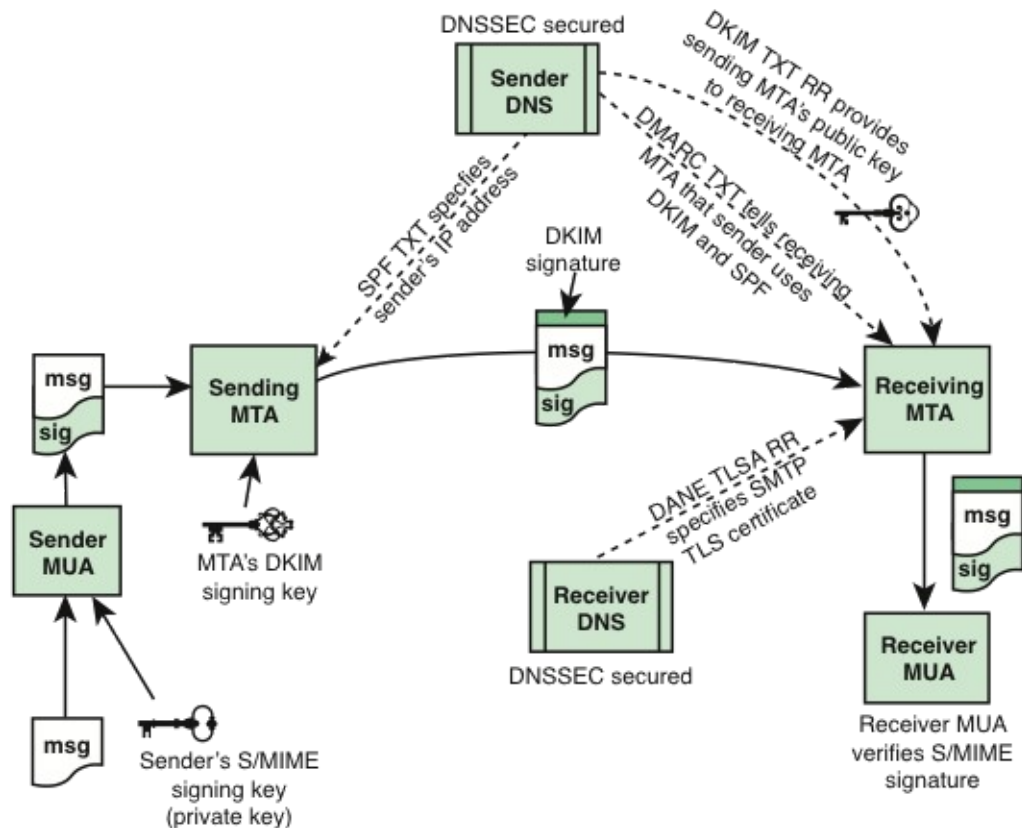
Threat	Impact on Purported Sender	Impact on Receiver	Mitigation
Email sent by unauthorized MTA in enterprise (e.g. malware botnet)	Loss of reputation, valid email from enterprise may be blocked as possible spam/phishing attack.	UBE and/or email containing malicious links may be delivered into user inboxes	Deployment of domain-based authentication techniques. Use of digital signatures over email.
Email message sent using spoofed or unregistered sending domain	Loss of reputation, valid email from enterprise may be blocked as possible spam/phishing attack.	UBE and/or email containing malicious links may be delivered into user inboxes	Deployment of domain-based authentication techniques. Use of digital signatures over email.
Email message sent using forged sending address or email address (i.e. phishing, spear phishing)	Loss of reputation, valid email from enterprise may be blocked as possible spam/phishing attack.	UBE and/or email containing malicious links may be delivered. Users may inadvertently divulge sensitive information or PII.	Deployment of domain-based authentication techniques. Use of digital signatures over email.
Email modified in transit	Leak of sensitive information or PII.	Leak of sensitive information, altered message may contain malicious information	Use of TLS to encrypt email transfer between server. Use of end-to-end email encryption.
Disclosure of sensitive information (e.g. PII) via monitoring and capturing of email traffic	Leak of sensitive information or PII.	Leak of sensitive information, altered message may contain malicious information	Use of TLS to encrypt email transfer between server. Use of end-to-end email encryption.
Unsolicited Bulk Email (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.



See  
<https://doi.org/10.6028/NIST.SP.800-177r1>

- **STARTTLS:**
- **S/MIME:**
- **DNS Security Extensions (DNSSEC):**
  - **1) DNS-based Authentication of Named Entities (DANE):** Is designed to overcome problems in the certificate authority (CA) system by providing an alternative channel for authenticating public keys based on DNSSEC, with the result that the same trust relationships used to certify IP addresses are used to certify servers operating on those addresses.
  - **2) Sender Policy Framework (SPF):** Uses the Domain Name System (DNS) to allow domain owners to create records that associate the domain name with a specific IP address range of authorized message senders. It is a simple matter for receivers to check the SPF TXT record in the DNS to confirm that the purported sender of a message is permitted to use that source address and reject mail that does not come from an authorized IP address.
  - **3) DomainKeys Identified Mail (DKIM):** Enables an MTA to sign selected headers and the body of a message. This validates the source domain of the mail and provides message body integrity.
  - **4) Domain-based Message Authentication, Reporting, and Conformance (DMARC):** Lets senders know the proportionate effectiveness of their SPF and DKIM policies, and signals to receivers what action should be taken in various individual and bulk attack scenarios.





DANE = DNS-based Authentication of Named Entities  
 DKIM = DomainKeys Identified Mail  
 DMARC = Domain-based Message Authentication, Reporting, and Conformance  
 DNSSEC = Domain Name System Security Extensions  
 SPF = Sender Policy Framework  
 S/MIME = Secure Multi-Purpose Internet Mail Extensions  
 TLSA RR = Transport Layer Security Authentication Resource Record

**Figure 8.4 The Interrelationship of DNSSEC, SPF, DKIM, DMARC, DANE, and S/MIME for Assuring Message Authenticity and Integrity**

# s/mime

Secure/  
Multipurpose  
Internet Mail  
Extension

S/MIME is a  
complex capability  
that is defined in a  
number of  
documents

Is a security  
enhancement to  
the MIME Internet  
e-mail format  
standard based on  
technology from  
RSA Data Security

# Table 8.4

## Summary of S/MIME 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.

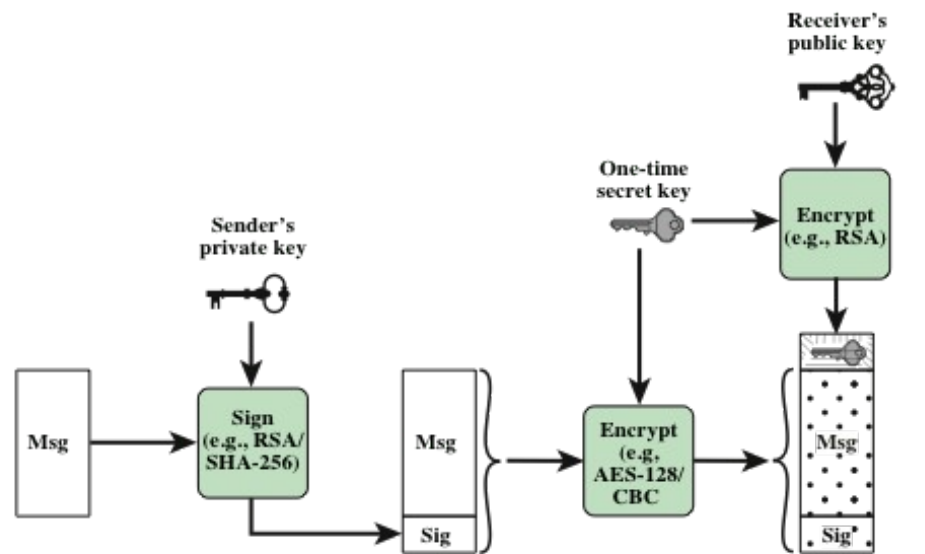
# authentication

- Authentication is provided by means of a digital signature
  - Most commonly RSA with SHA-256 is used
    - The sender creates a message
    - SHA-256 is used to generate a 256-bit message digest of the message
    - The message digest is encrypted with RSA using the sender's private key, and the result is appended to the message; also appended is identifying information for the signer, which will enable the receiver to retrieve the signer's public-key
    - The receiver uses RSA with the sender's public key to decrypt and recover the message digest
    - The receiver generates a new message digest for the message and compares it with the decrypted hash code; if the two match, the message is accepted as authentic

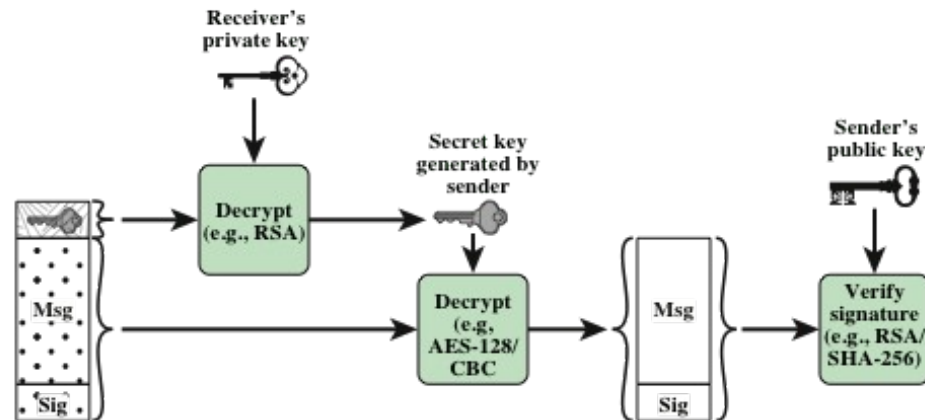
# confidentiality

- S/MIME provides confidentiality by encrypting messages
- Most commonly AES with a 128-bit key is used, with the cipher block chaining (CBC) mode
- The key itself is also encrypted, typically with RSA
- In S/MIME each symmetric key, referred to as a content-encryption key, is used only once
- To protect the key, it is encrypted with the receiver's public key
- Sequence:
  - The sender generates a message and a random 128-bit number to be used as a content-encryption key for this message only
  - The message is encrypted using the content-encryption key
  - The content-encryption key is encrypted with RSA using the recipient's public key and is attached to the message
  - The receiver uses RSA with its private key to decrypt and recover the content-encryption key
  - The content-encryption key is used to decrypt the message





(a) Sender signs, then encrypts message

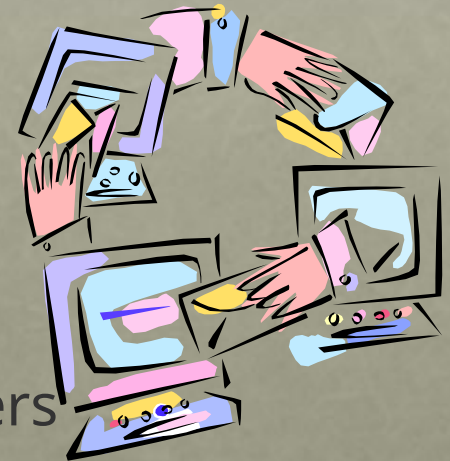


(b) Receiver decrypts message, then verifies sender's signature

Figure 8.5 Simplified S/MIME Functional Flow

# E-mail compatibility

- Many electronic mail systems only permit the use of blocks consisting of ASCII text
- To accommodate this restriction, S/MIME provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII characters
  - A process referred to as 7-bit encoding
- The scheme typically used for this purpose is Base64 conversion
  - Each group of three octets of binary data is mapped into four ASCII characters



# compression

- S/MIME also offers the ability to compress a message
  - This has the benefit of saving space both for e-mail transmission and for file storage
- Compression can be applied in any order with respect to the signing and message encryption operations
- RFC 5751 provides the following guidelines:
  - Compression of binary encoded encrypted data is discouraged, since it will not yield significant compression; Base64 encrypted data could very well benefit, however
  - If a lossy compression algorithm is used with signing, you will need to compress first, then sign

# s/Mime message content types

S/MIME uses the following message content types, which are defined in RFC 5652, Cryptographic Message Syntax

## Data

- Refers to the inner MIME-encoded message content, which may then be encapsulated in a SignedData, EnvelopedData, or CompressedData content type

## SignedData

- Used to apply a digital signature to a message

## EnvelopedData

- This consists of encrypted content of any type and encrypted-content encryption keys for one or more recipients

## CompressedData

- Used to apply data compression to a message

# Table 8.5

## Cryptographic Algorithms Used in S/MIME

Function	Requirement
Create a message digest to be used in forming a digital signature.	MUST support SHA-256 SHOULD support SHA-1 Receiver SHOULD support MD5 for backward compatibility
Use message digest to form a digital signature.	MUST support RSA with SHA-256 SHOULD support <ul style="list-style-type: none"> <li>— DSA with SHA-256</li> <li>— RSASSA-PSS with SHA-256</li> <li>— RSA with SHA-1</li> <li>— DSA with SHA-1</li> <li>— RSA with MD5</li> </ul>
Encrypt session key for transmission with a message.	MUST support RSA encryption SHOULD support <ul style="list-style-type: none"> <li>— RSAES-OAEP</li> <li>— Diffie-Hellman ephemeral-static mode</li> </ul>
Encrypt message for transmission with a one-time session key.	MUST support AES-128 with CBC SHOULD support <ul style="list-style-type: none"> <li>— AES-192 CBC and AES-256 CBC</li> <li>— Triple DES CBC</li> </ul>



# s/mime Content types – envelopedData

- The steps for preparing an envelopedData MIME entity are:
  - Generate a pseudorandom session key for a particular symmetric encryption algorithm (RC2/40 or triple DES)
  - For each recipient, encrypt the session key with the recipient's public RSA key
  - For each recipient, prepare a block known as *RecipientInfo* that contains an identifier of the recipient's public-key certificate
  - Encrypt the message content with the session key

# s/mime Content types – signedData

The steps for preparing a signedData MIME entity are as follows:

Select a message digest algorithm (SHA or MD5)

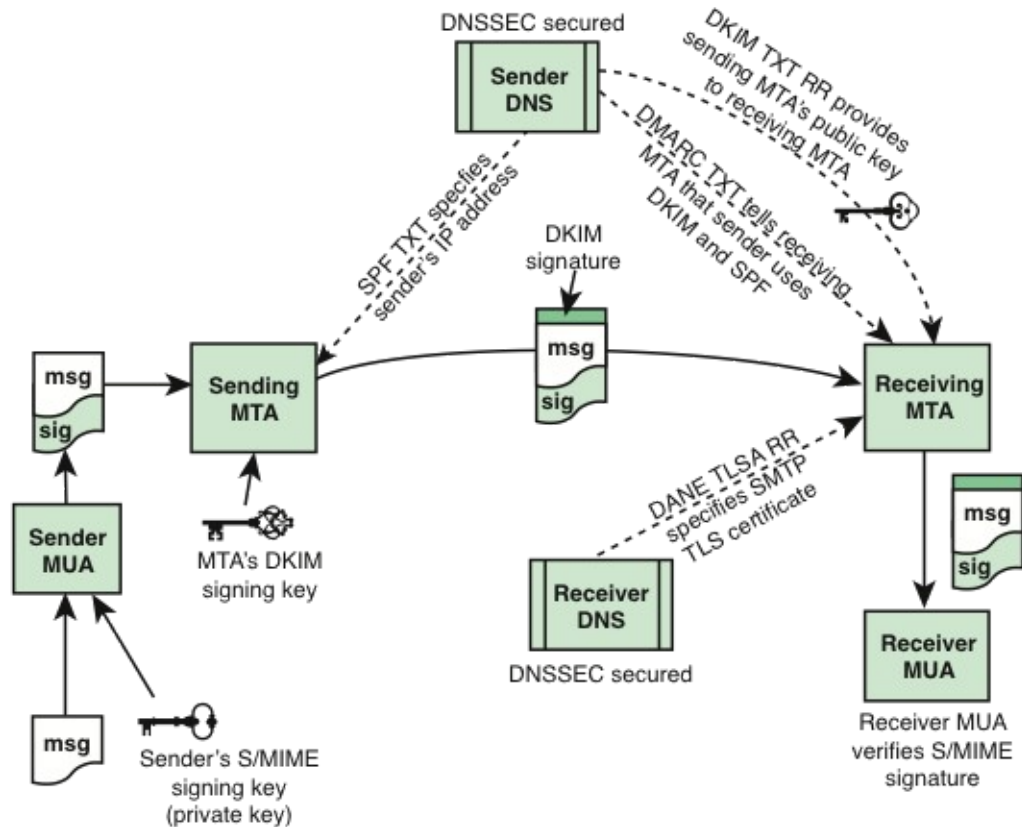
Compute the message digest (hash function) of the content to be signed

Encrypt the message digest with the signer's private key

Prepare a block known as *SignerInfo* that contains the signer's public-key certificate, an identifier of the message digest algorithm, an identifier of the algorithm used to encrypt the message digest, and the encrypted message digest

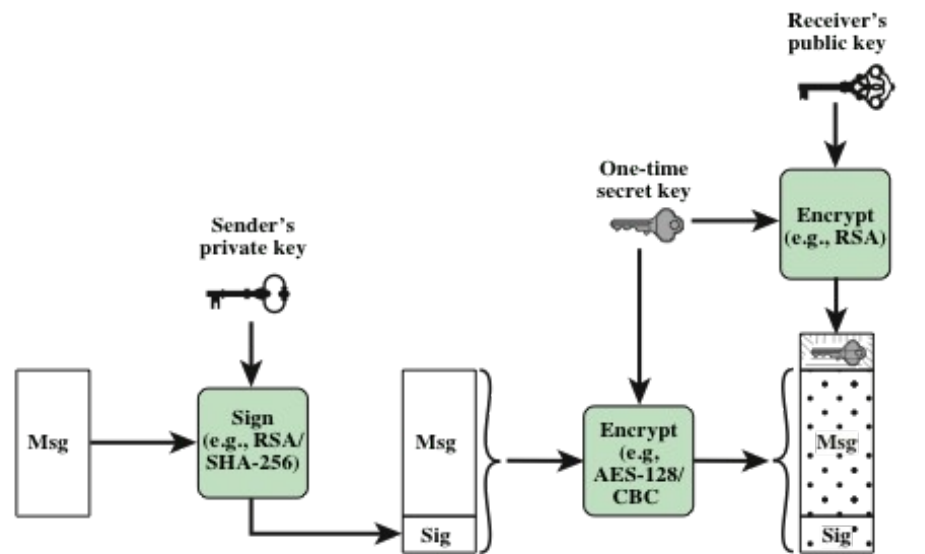
# S/mime certificate processing

- A S/MIME user has several key management functions to perform:
  - Key generation
    - The user or some related administrative utility MUST be capable of generating separate Diffie-Hellman and DSS key pairs and SHOULD be capable of generating RSA key pairs
    - Each key pair MUST be generated from a good source of nondeterministic random input and be protected in a secure fashion
    - A user agent SHOULD generate RSA key pairs with a length in the range of 768 to 1024 bits and MUST NOT generate a length of less than 512 bits
  - Registration
    - A user's public key must be registered with a certification authority in order to receive an X.509 public-key certificate
  - Certificate storage and retrieval
    - A user requires access to a local list of certificates in order to verify incoming signatures and to encrypt outgoing messages
      - Such a list could be maintained by the user or by some local administrative entity on behalf of a number of users

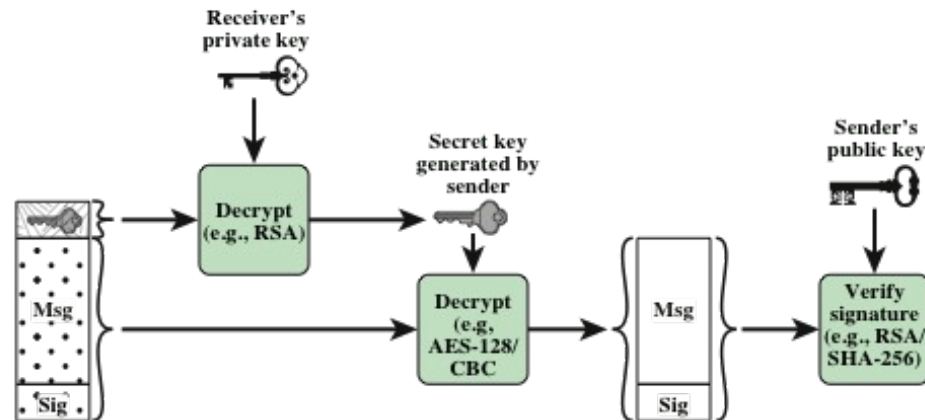


DANE = DNS-based Authentication of Named Entities  
 DKIM = DomainKeys Identified Mail  
 DMARC = Domain-based Message Authentication, Reporting, and Conformance  
 DNSSEC = Domain Name System Security Extensions  
 SPF = Sender Policy Framework  
 S/MIME = Secure Multi-Purpose Internet Mail Extensions  
 TLSA RR = Transport Layer Security Authentication Resource Record

**Figure 8.4 The Interrelationship of DNSSEC, SPF, DKIM, DMARC, DANE, and S/MIME for Assuring Message Authenticity and Integrity**



(a) Sender signs, then encrypts message

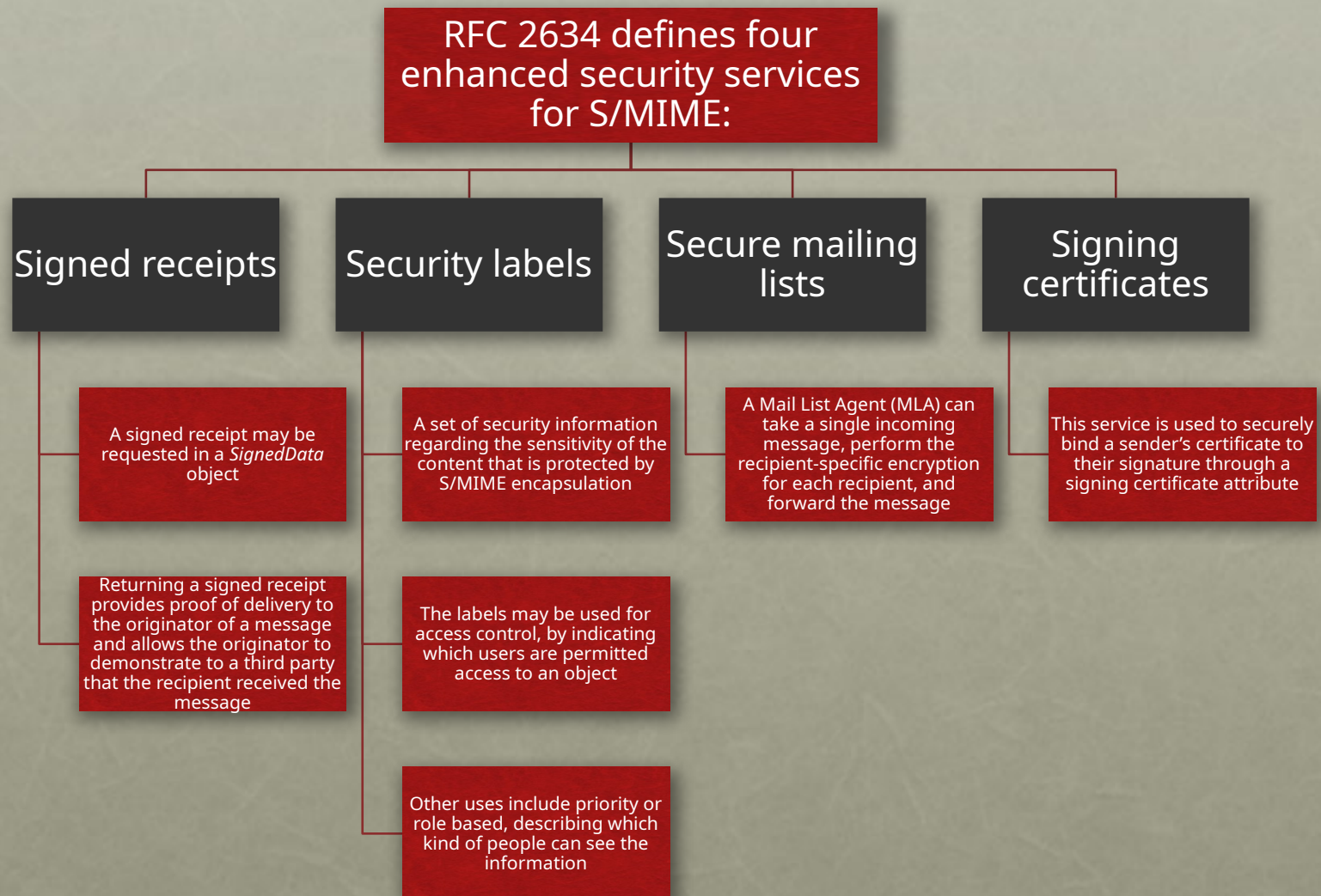


(b) Receiver decrypts message, then verifies sender's signature

Figure 8.5 Simplified S/MIME Functional Flow



# Enhanced security services

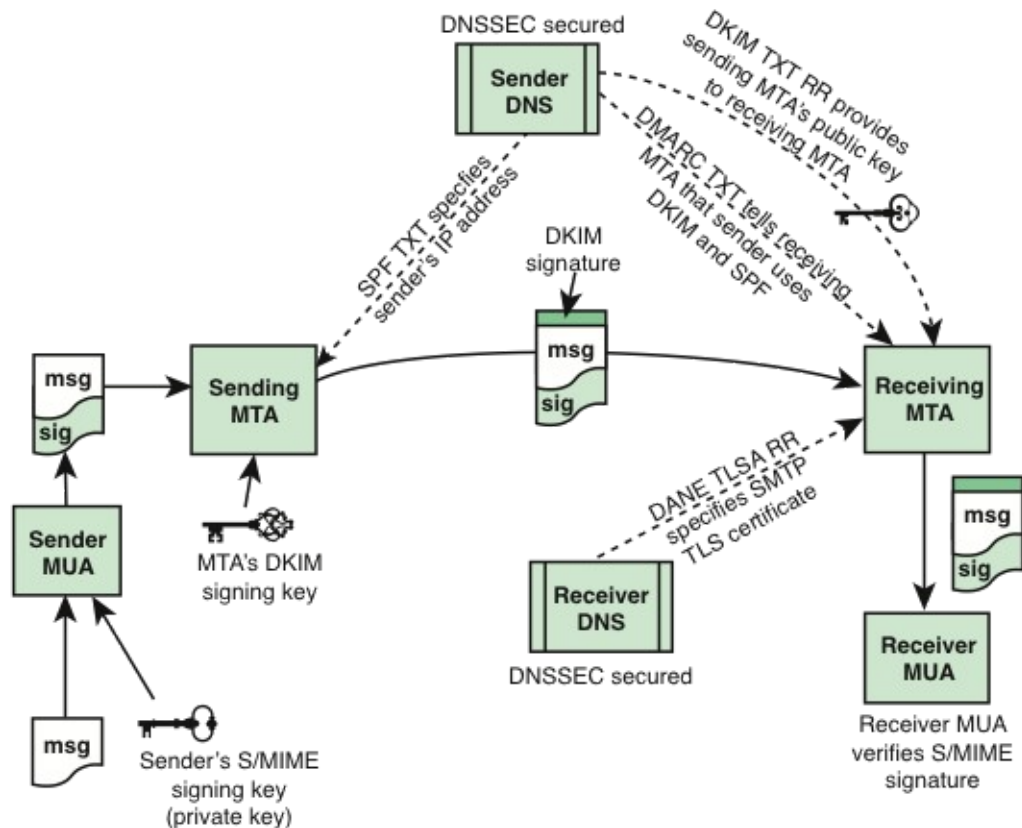


# S/MIME vs PEC

- <https://archiviando.wordpress.com/2015/02/23/la-pec-un-prodotto-made-in-italy/>
- [https://it.wikipedia.org/wiki/Posta\\_elettronica\\_certificata](https://it.wikipedia.org/wiki/Posta_elettronica_certificata)
- <https://news.saferbytes.it/approfondimenti/2010/05/pec-e-smime-equivalenti-o-complementari/>
- [https://www.ilsoftware.it/articoli.asp?tag=Posta-elettronica-certificata-e-certificati-s-mime-differenze-e-funzionamento\\_4957](https://www.ilsoftware.it/articoli.asp?tag=Posta-elettronica-certificata-e-certificati-s-mime-differenze-e-funzionamento_4957)
- PEC non garantisce a priori l'integrità del messaggio. La PEC garantisce solamente che l'e-mail inviata dal mittente non venga alterata durante il trasferimento da provider a provider, ma non garantisce che l'e-mail scritta dal mittente non sia stata alterata nel percorso tra il mittente e il provider del mittente.

# S/MIME vs PEC

- 7 maggio 2010
  - Qualsiasi utente può disporre di un proprio certificato digitale S/MIME, grazie al quale può firmare le proprie e-mail digitalmente. Firmando digitalmente la propria e-mail, questa e-mail assumerà valore legale.
  - Infatti, secondo l'Art. 21 del D.Lgs n°82 del 7 Marzo 2005, "Il documento informatico, sottoscritto con firma digitale o con un altro tipo di firma elettronica qualificata, ha l'efficacia prevista dall'articolo 2702 del codice civile".
  - Essendo lo standard S/MIME un tipo di firma elettronica qualificata, l'e-mail assume validità legale. Ciò significa che, inviando una e-mail con una firma digitale, quella e-mail potrà essere utilizzata in tribunale se ci dovessero essere eventuali cause legali.



DANE = DNS-based Authentication of Named Entities  
 DKIM = DomainKeys Identified Mail  
 DMARC = Domain-based Message Authentication, Reporting, and Conformance  
 DNSSEC = Domain Name System Security Extensions  
 SPF = Sender Policy Framework  
 S/MIME = Secure Multi-Purpose Internet Mail Extensions  
 TLSA RR = Transport Layer Security Authentication Resource Record

**Figure 8.4 The Interrelationship of DNSSEC, SPF, DKIM, DMARC, DANE, and S/MIME for Assuring Message Authenticity and Integrity**



See  
<https://doi.org/10.6028/NIST.SP.800-177r1>

- **STARTTLS:**
- **S/MIME:**
- **DNS Security Extensions (DNSSEC):**
  - **1) DNS-based Authentication of Named Entities (DANE):** Is designed to overcome problems in the certificate authority (CA) system by providing an alternative channel for authenticating public keys based on DNSSEC, with the result that the same trust relationships used to certify IP addresses are used to certify servers operating on those addresses.
  - **2) Sender Policy Framework (SPF):** Uses the Domain Name System (DNS) to allow domain owners to create records that associate the domain name with a specific IP address range of authorized message senders. It is a simple matter for receivers to check the SPF TXT record in the DNS to confirm that the purported sender of a message is permitted to use that source address and reject mail that does not come from an authorized IP address.
  - **3) DomainKeys Identified Mail (DKIM):** Enables an MTA to sign selected headers and the body of a message. This validates the source domain of the mail and provides message body integrity.
  - **4) Domain-based Message Authentication, Reporting, and Conformance (DMARC):** Lets senders know the proportionate effectiveness of their SPF and DKIM policies, and signals to receivers what action should be taken in various individual and bulk attack scenarios.



# Domain name system (dns)


- A directory lookup service that provides a mapping between the name of a host on the Internet and its numeric IP address
- Is essential to the functioning of the Internet
- Is used by MUAs and MTAs to find the address of the next hop server for mail delivery
- Comprised of four elements:
  - Domain name space
    - DNS uses a tree-structured name space to identify resources on the Internet
  - DNS database
    - Conceptually, each node and leaf in the name space tree structure names a set of information that is contained in resource record. The collection of all RRs is organized into a distributed database
  - Name servers
    - These are server programs that hold information about a portion of the domain name tree structure and the associated RRs
  - Resolvers
    - These are programs that extract information from name servers in response to client requests. A typical client request is for an IP address corresponding to a given domain name

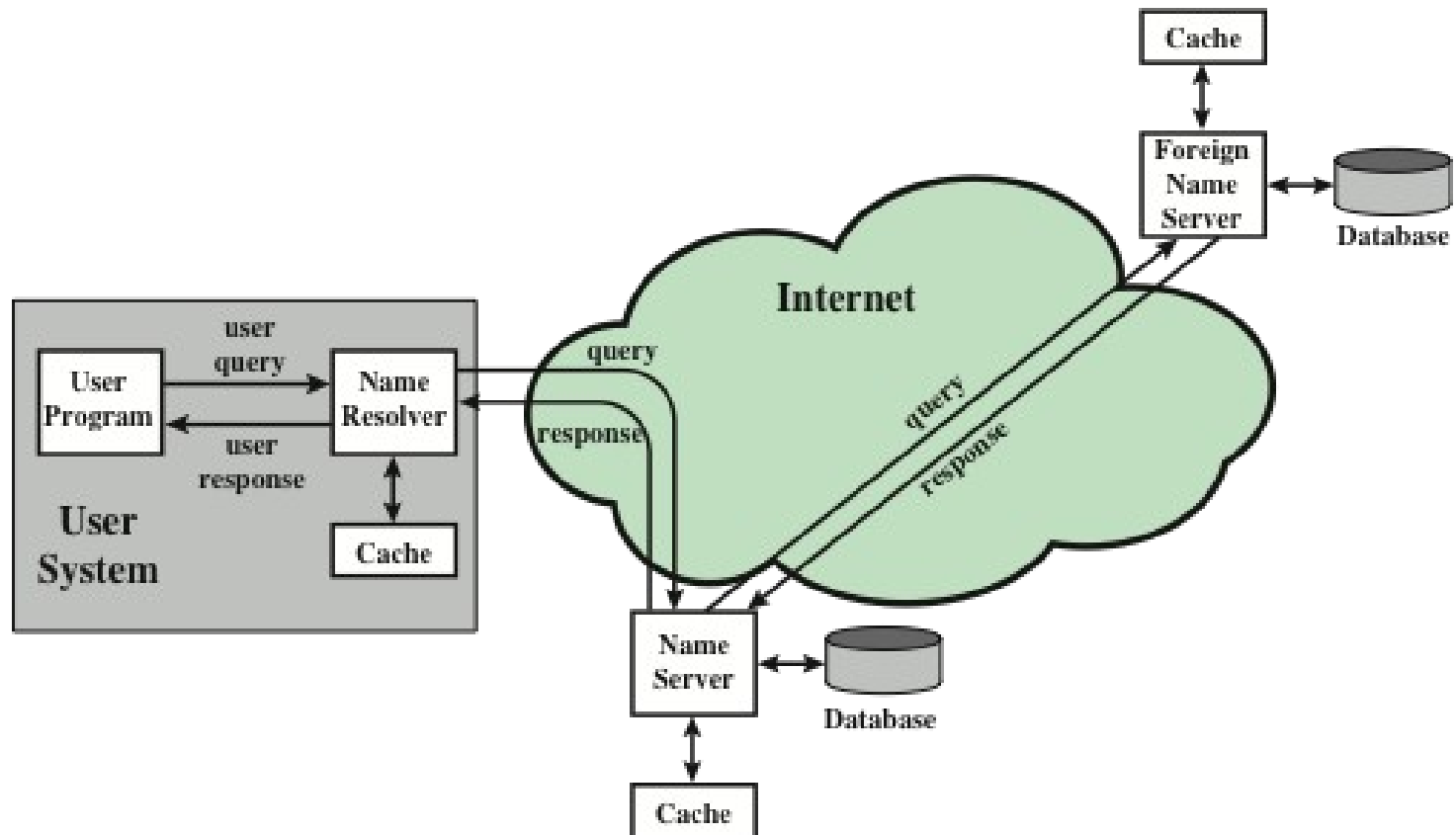
# Dns database

- DNS is based on a hierarchical database containing resource records (RRs) that include the name, IP address, and other information about hosts
- Key features of the database:
  - Variable-depth hierarchy for names
    - DNS allows essentially unlimited levels and uses the period (.) as the level delimiter in printed names
  - Distributed database
    - The database resides in DNS servers scattered throughout the Internet
  - Distribution controlled by the database
    - The DNS database is divided into thousands of separately managed zones, which are managed by separate administrators

# Table 8.6

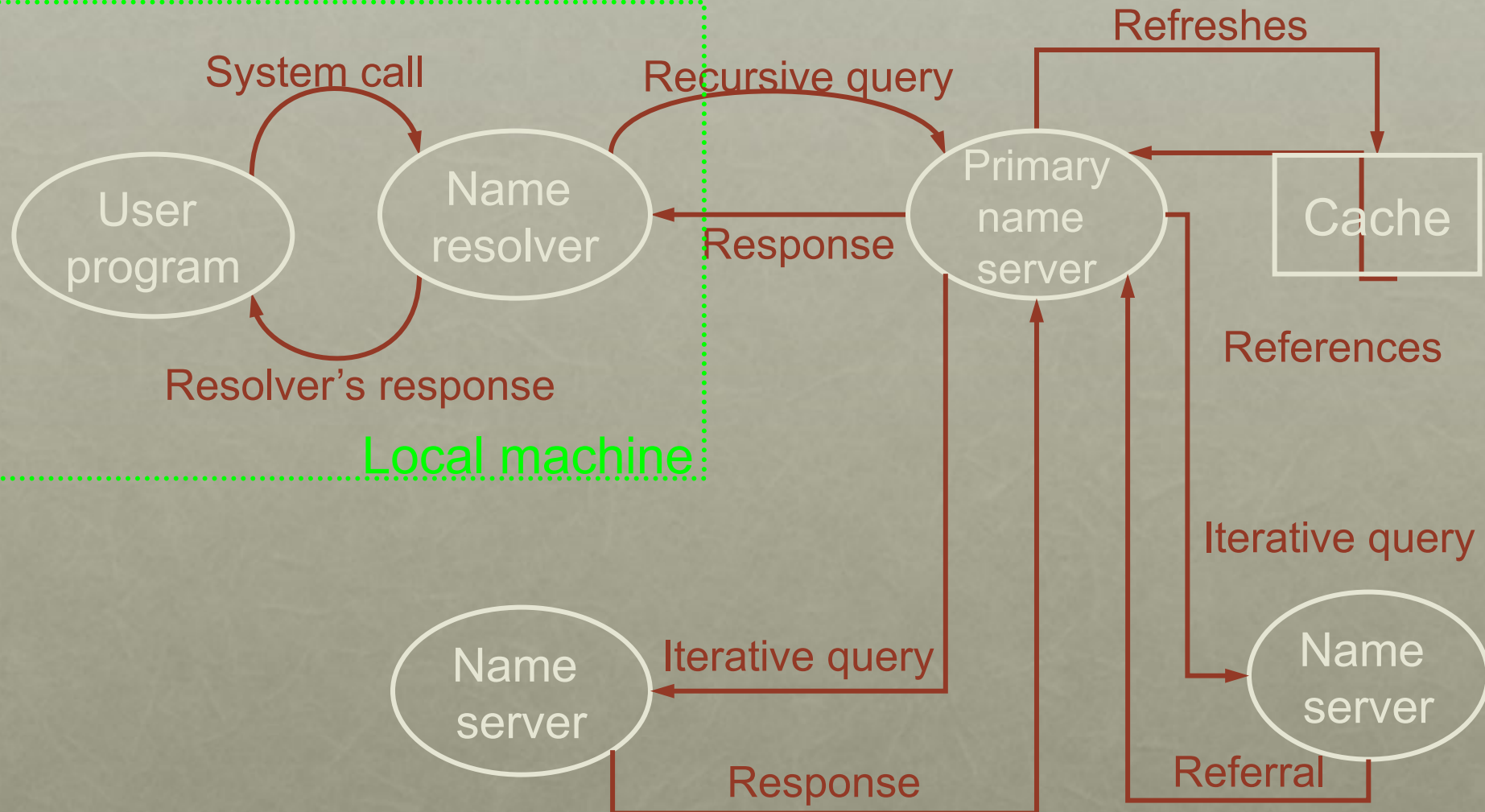
## Resource Record Types

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.



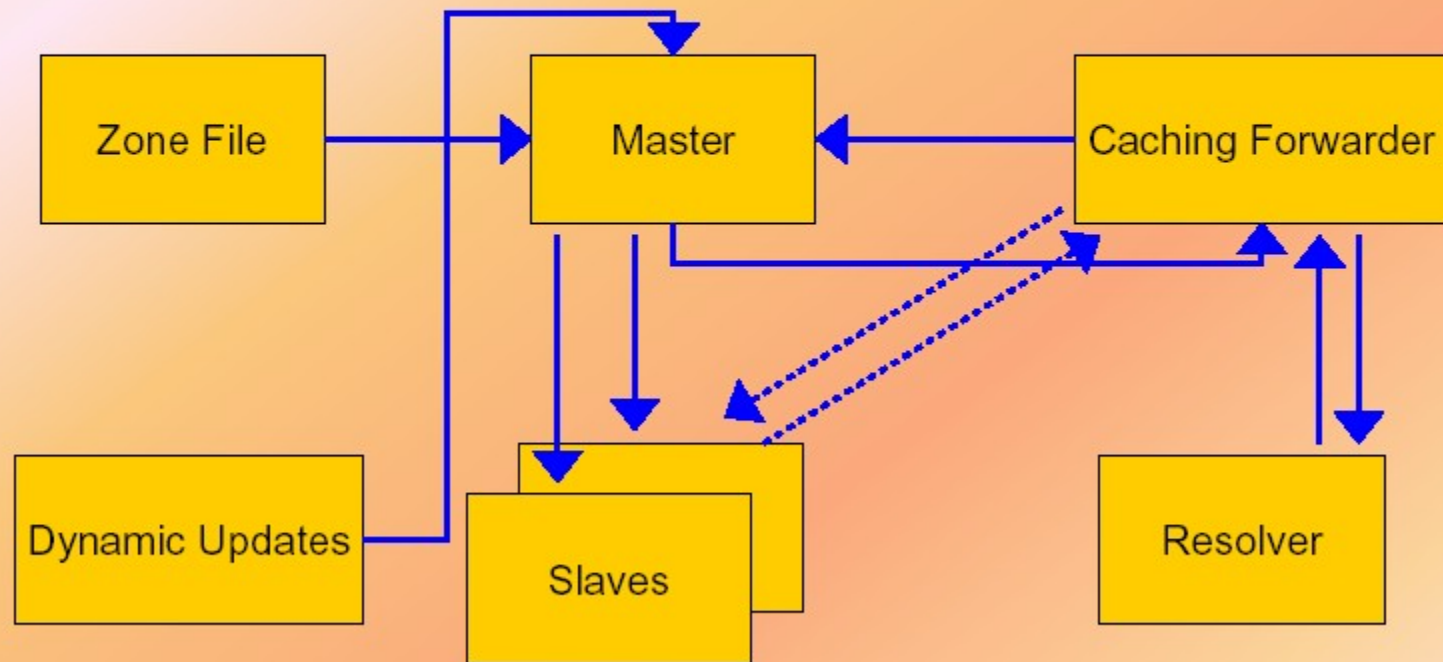
**Figure 8.6 DNS Name Resolution**

# Processo di risoluzione dei nomi

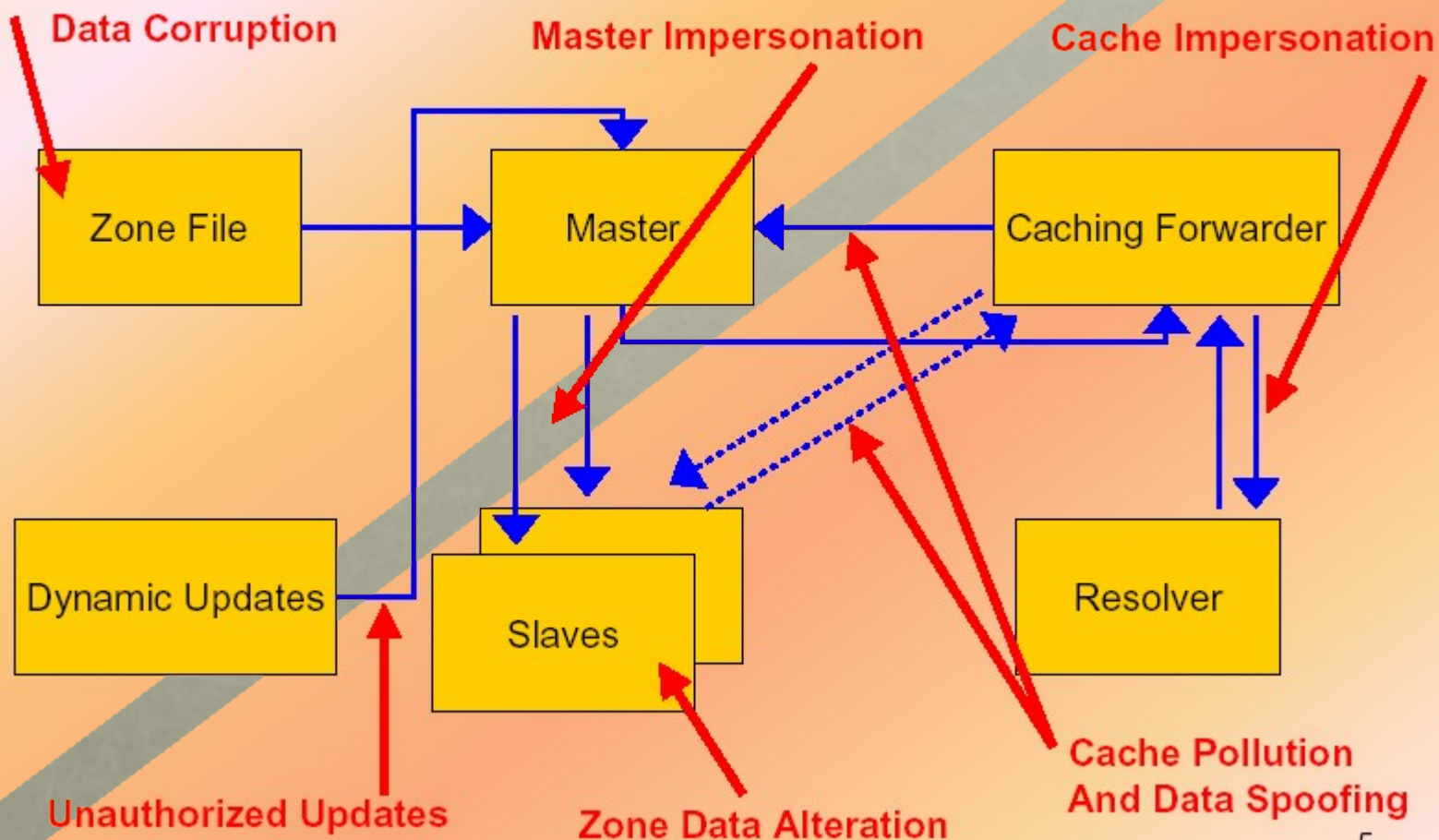




# DNS: Data Flow



# DNS: Vulnerabilità



# Dns security extensions

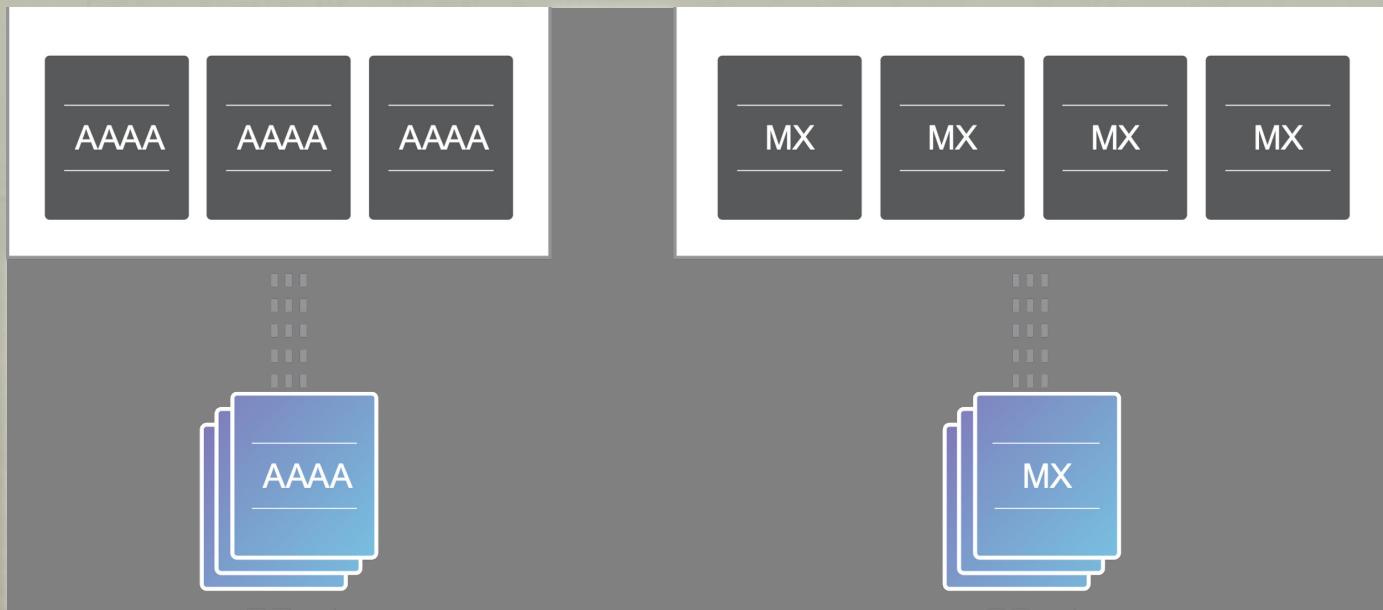
- DNSSEC:
  - Provides end-to-end protection through the use of digital signatures that are created by responding zone administrators and verified by a recipient's resolver software
  - Avoids the need to trust intermediate name servers and resolvers that cache or route the DNS records originating from the responding zone administrator before they reach the source of the query
  - Consists of a set of new resource record types and modifications to the existing DNS protocol
- Defined in:
  - RFC 4033, *DNS Security Introduction and Requirements*
  - RFC 4034, *Resource Records for the DNS Security Extensions*
  - RFC 4035, *Protocol Modifications for the DNS Security Extensions*

- DNSSEC crea un sistema di nomi di dominio sicuro aggiungendo le firme crittografiche ai record DNS esistenti. Queste firme digitali vengono memorizzate nei nameserver DNS insieme a tipi di record più comuni, come A, AAAA, MX, CNAME, ecc. Controllando la firma associata, è possibile verificare che il record DNS richiesto provenga dal suo nameserver autoritativo e che non sia stato modificato durante il percorso, a differenza di un record falso iniettato in un attacco man-in-the-middle.



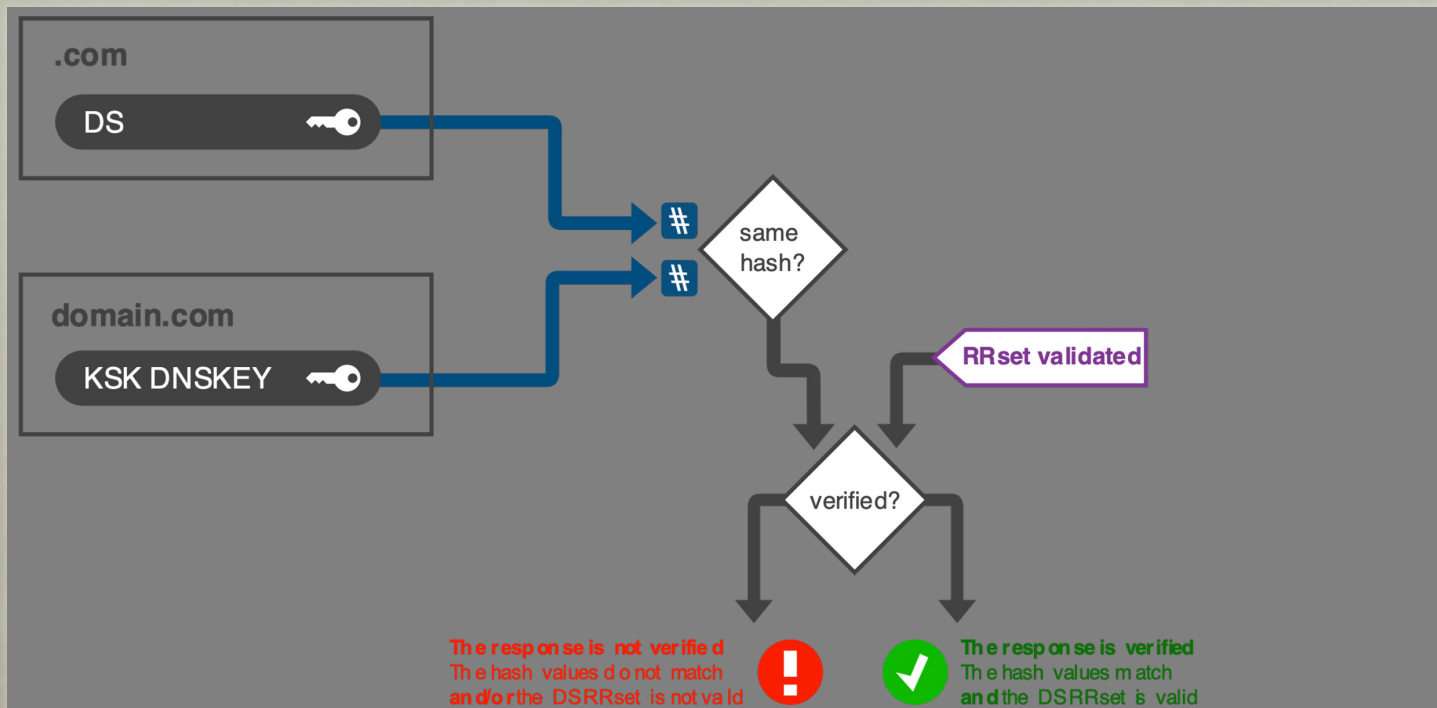
- Per facilitare la convalida della firma, DNSSEC aggiunge alcuni nuovi tipi di record DNS:
- **RRSIG** - Contiene una firma crittografica
- **DNSKEY** - Contiene una chiave di firma pubblica
- **DS** - Contiene l'hash di un record DNSKEY
- **NSEC** e **NSEC3** - Per il denial-of-existence esplicito di un record DNS
- **CDNSKEY** e **CDS** - Per una zona figlio che richiede aggiornamenti ai record DS nella zona genitore.





## RRSet

- Il primo passo per proteggere un'area con DNSSEC consiste nel raggruppare tutti i record dello stesso tipo in un insieme di dati detto RRSet (Resource Record Set).
- In realtà, è questo RRSet completo che viene firmato digitalmente, a differenza dei singoli record DNS.



# Authenticity of DNS records

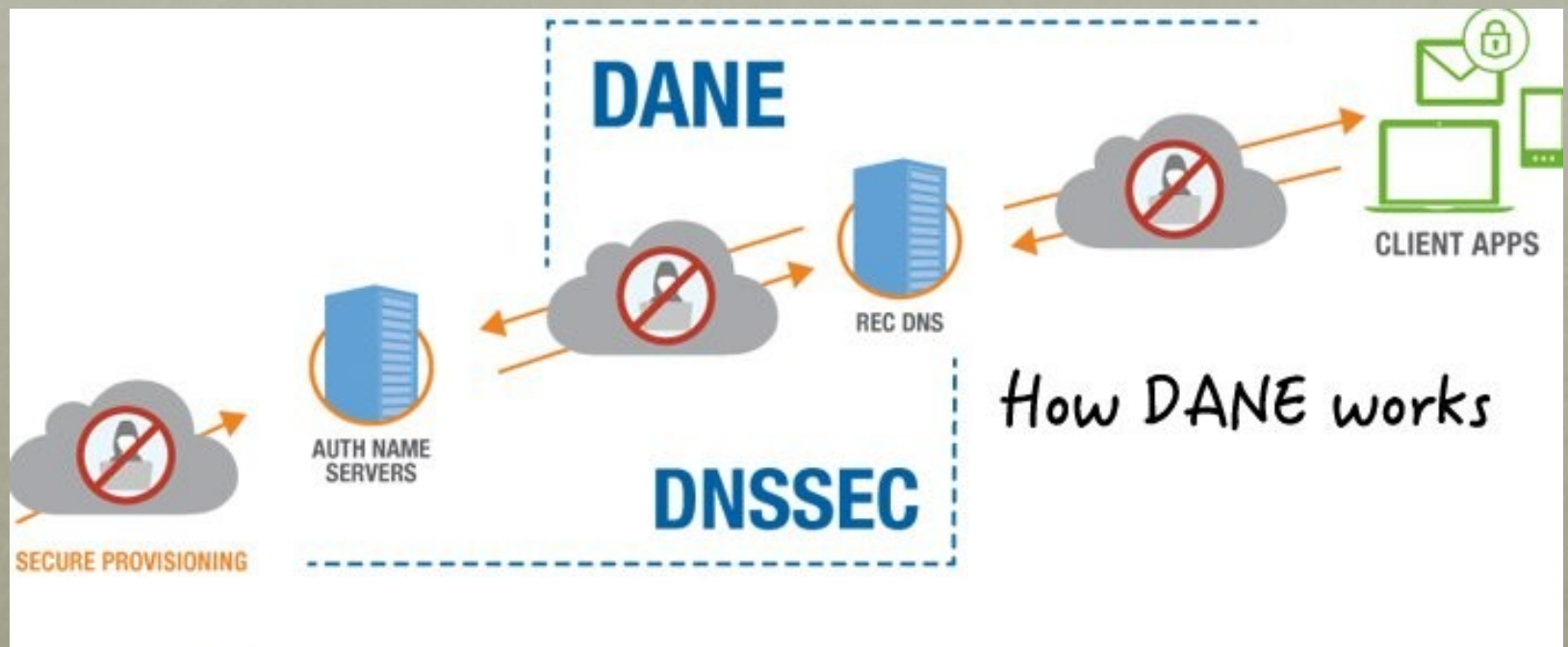
1. get the public Key Signing Key of the zone using DNS query (DNSKEY KSK)
2. get the signature of the Zone Signing Key public key record using DNS (RRSIG DNSKEY ZSK)
3. verify the signature of the ZSK with the public key of the KSK
4. from the KSK public key, build the hash that creates the DS record
5. get the DS record from the upper zone using DNS
6. compare DS contained in upper zone and hash, should be the same

# Dettagli

- <https://www.cloudflare.com/it-it/dns/dnssec/how-dnssec-works/>
- Test
- <https://dnssec-analyzer.verisignlabs.com>

# 1) Dns-based authentication of named entities (dane)

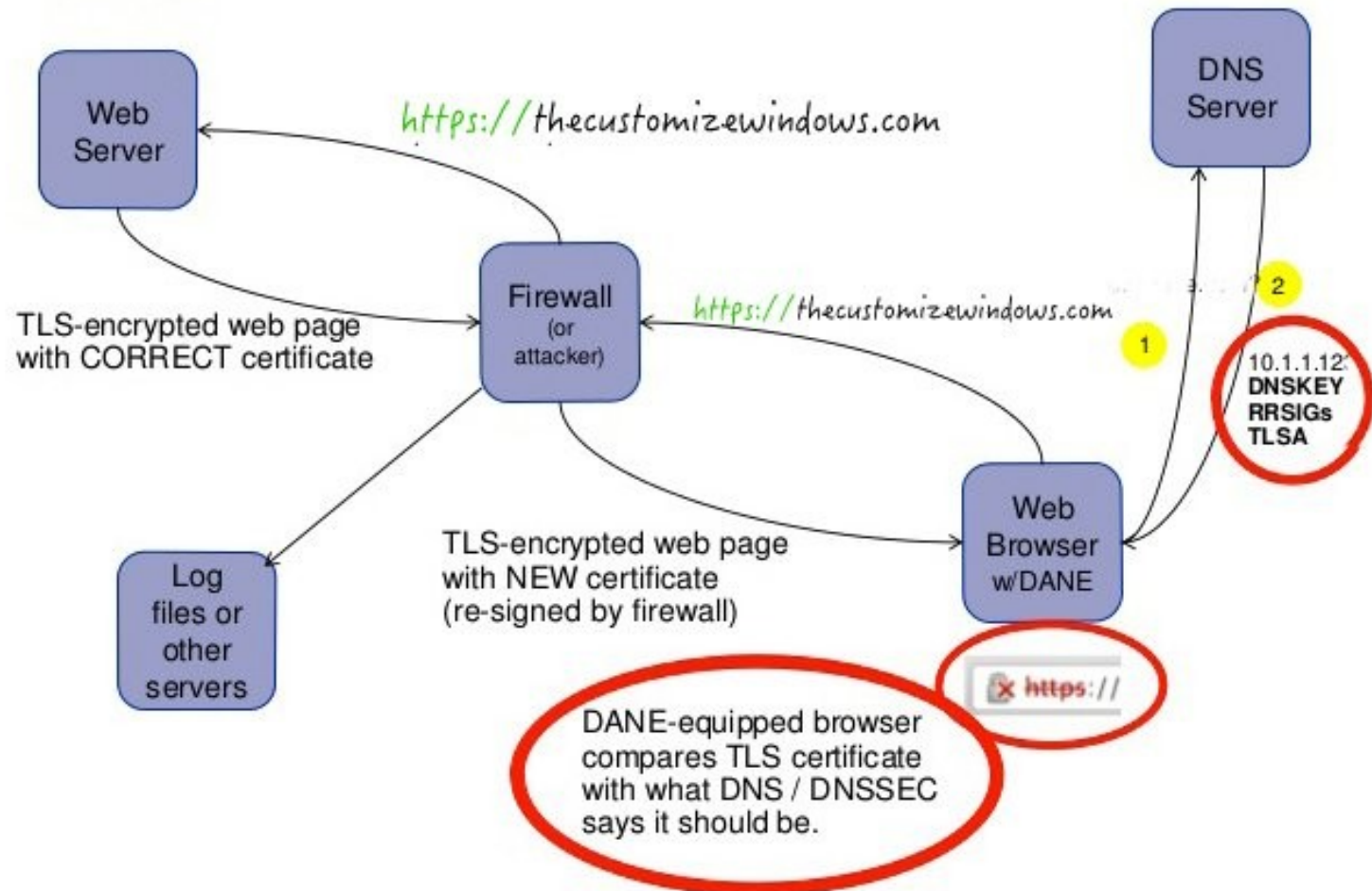
- DANE is a protocol to allow X.509 certificates, commonly used for Transport Layer Security (TLS), to be bound to DNS names using DNSSEC
- It is proposed in RFC 6698 as a way to authenticate TLS client and server entities without a certificate authority (CA)
- The purpose of DANE is to replace reliance on the security of the CA system with reliance on the security provided by DNSSEC
- DANE defines a new DNS record type, TLSA, that can be used for a secure method of authenticating SSL/TLS certificates



How DANE works



# DANE



port, transport proto &  
server domain name

TLSA rrtype


`_443._tcp.www.example.com. IN TLSA (`  
`0 0 1 d2abde240d7cd3ee6b4b28c54df034b9`  
`7983a1d16e8a410e4561cb106618e971 )`

usage  
selector  
matching type  
certificate association data

thecustomizewindows.com

Google Tran... Google Tran...

Client Area ~... ze Windows Cus

 **https://thecustomizewindows.com**  
**Certificate corresponds to TLSA**

The remote server certificate for this domain name was verified by DANE protocol. The certificate corresponds to TLSA record which is secured by DNSSEC technology.

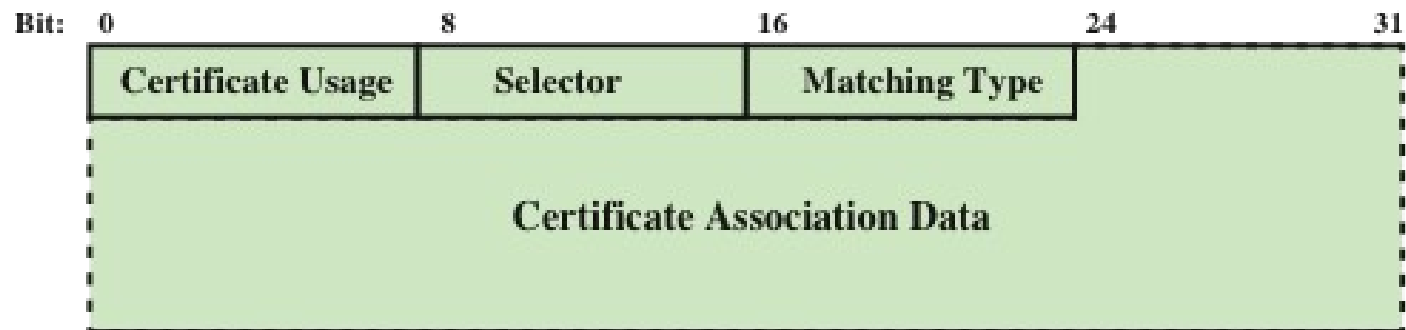
Home A

THE CUST G VIRTUALIZATION COM

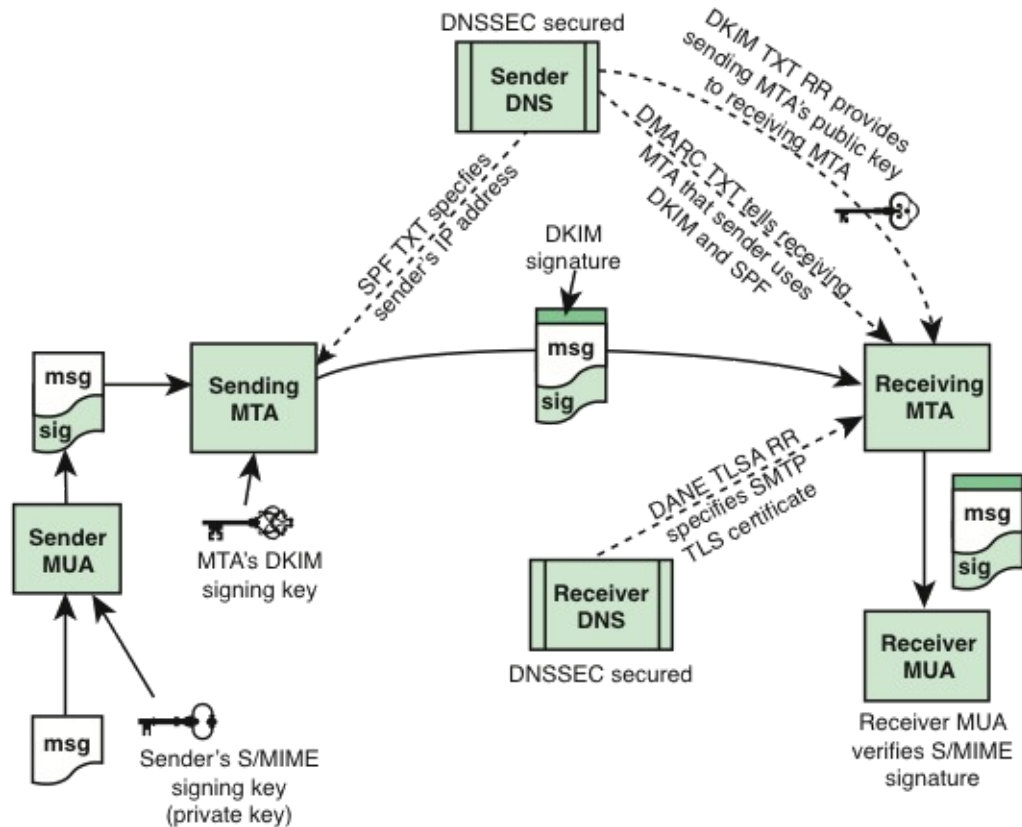
# check

- <https://www.huque.com/bin/danecheck>
- <https://www.internetsociety.org/resources/deploy360/dane-test-sites/>
- <https://dane.sys4.de/smtp/openssl.org>
- <https://check.sidnlabs.nl/dane/>
- <https://www.cloudns.net/wiki/article/342/>

Host	Type	Points to:	TTL
_port._protocol.host.domain.com	TLSA	0 0 0 000000000000000000000000	1 Hour



**Figure 8.7 TLSA RR Transmission Format**



DANE = DNS-based Authentication of Named Entities  
 DKIM = DomainKeys Identified Mail  
 DMARC = Domain-based Message Authentication, Reporting, and Conformance  
 DNSSEC = Domain Name System Security Extensions  
 SPF = Sender Policy Framework  
 S/MIME = Secure Multi-Purpose Internet Mail Extensions  
 TLSA RR = Transport Layer Security Authentication Resource Record

**Figure 8.4 The Interrelationship of DNSSEC, SPF, DKIM, DMARC, DANE, and S/MIME for Assuring Message Authenticity and Integrity**



## 2) Sender policy framework (spf)

- SPF is the standardized way for a sending domain to identify and assert the mail senders for a given domain
- Addresses the problem of any host being able to use any domain name for each of the various identifiers in the mail header, not just the domain name where the host is located
- Defined in RFC 7208
- SPF works by checking a sender's IP address against the policy encoded in any SPF record found at the sending domain

```
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
. . .
```

**Figure 8.8 Example in which SMTP Envelope Header Does Not Match Message Header**

- > set querytype=SPF
- > unipg.it
- Server: 192.168.1.1
- Address: 192.168.1.1#53
- Non-authoritative answer:
- unipg.it rdata\_99 = "v=spf1 include:spf.protection.outlook.com include:\_spf.cineca.it " " ip4:141.250.196.0/22 ip4:90.147.160.68/32 ip4:141.250.1.0/24 ip4:141.250.249.224/27 ip4:95.172.90.156 ip4:31.186.254.152 ip4:141.250.84.198/31 ip4:141.250.84.200/32 ip4:95.172.90.143 ip4:130.186.7.110 mx -all"

- > set querytype=TXT
- > unipg.it
- Server: 192.168.1.1
- Address: 192.168.1.1#53
- Non-authoritative answer:
- unipg.it text = "v=spf1 include:spf.protection.outlook.com include:\_spf.cineca.it " " ip4:141.250.196.0/22 ip4:90.147.160.68/32 ip4:141.250.1.0/24 ip4:141.250.249.224/27 ip4:95.172.90.156 ip4:31.186.254.152 ip4:141.250.84.198/31 ip4:141.250.84.200/32 ip4:95.172.90.143 ip4:130.186.7.110 mx -all"

**Table 8.7 Common SPF Mechanisms and Modifiers****(a) SPF 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.

**(b) SPF Mechanism Modifiers**

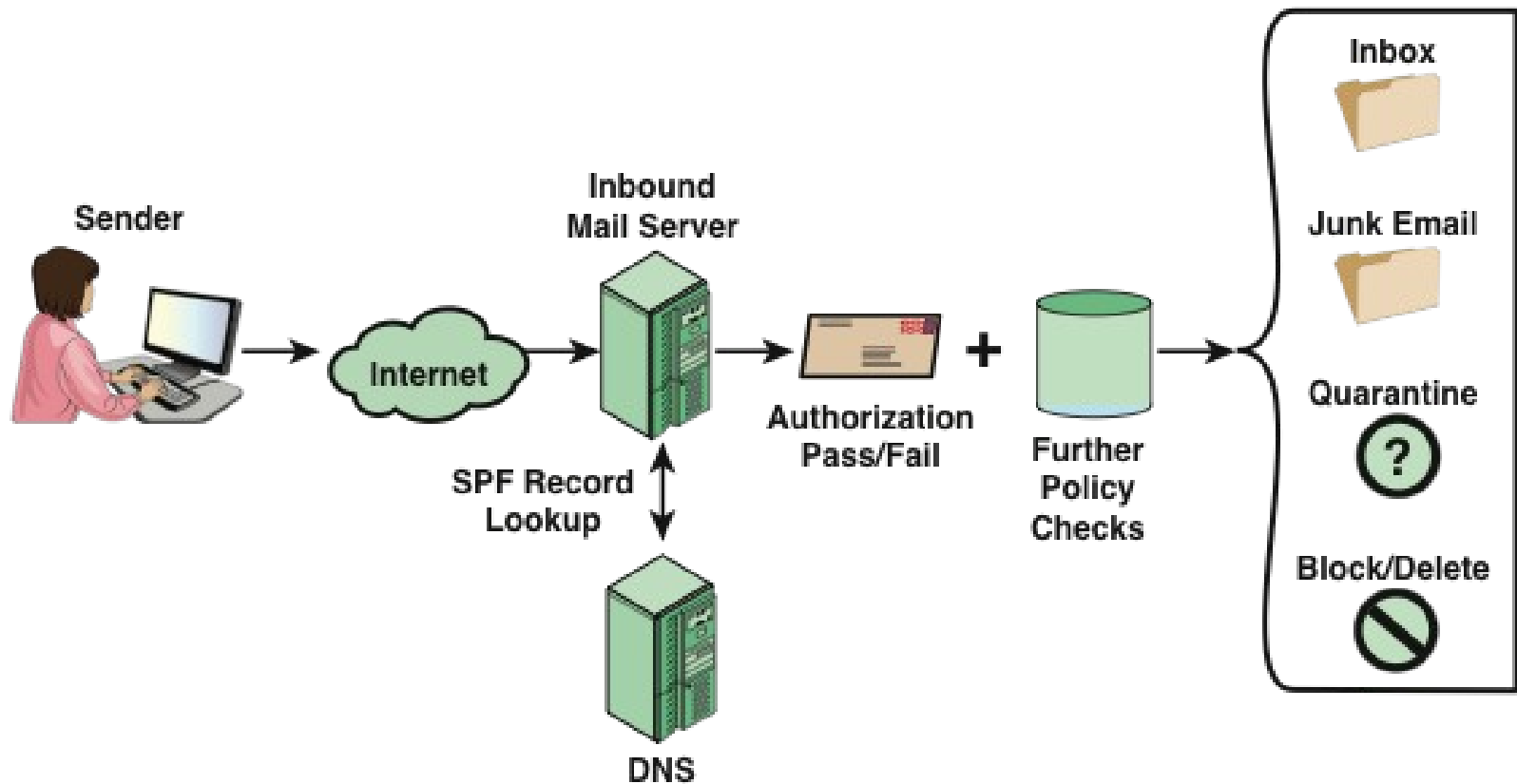
Modifier	Description
+	The given mechanism check must pass. This is the default mechanism and does not need to be explicitly listed.
–	The given mechanism is not allowed to send email on behalf of the domain.
~	The given mechanism is in transition and if an email is seen from the listed host/IP address, that 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).

(Table can be found on page 273 in the textbook)



# Spf lookups

- <https://mxtoolbox.com/spf.aspx>



**Figure 8.9 Sender Policy Framework Operation**

# When are these values used

The following table summarizes the two different values for the sender.

## Envelope From (RFC5321.MAIL FROM)

- Used by the SMTP server to generate NDR
- Used by **SPF filter** to determine if it came from the designated IP address.

## Header From (RFC5322.FROM)

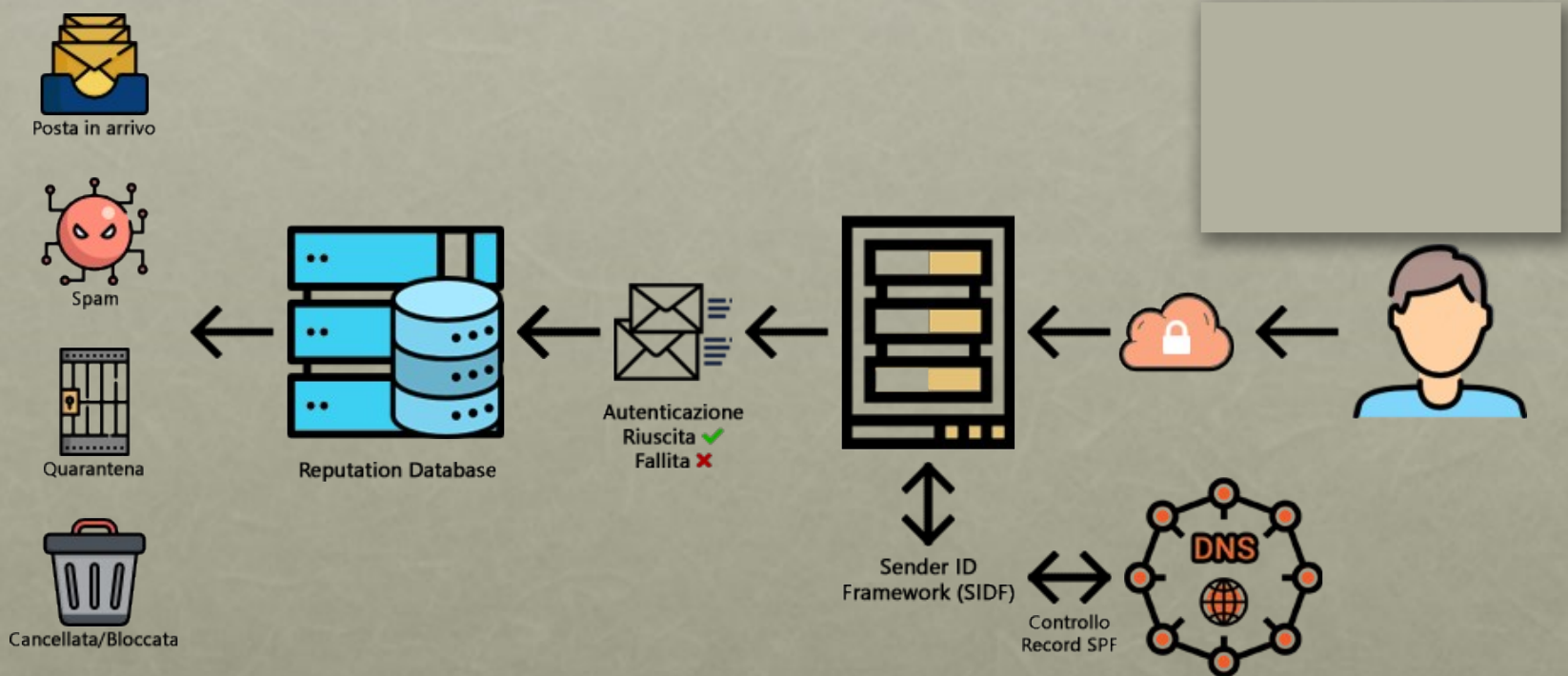
- Used by the email client to display information in the From field.
- Used by **DMARC filter** to confirm if the message is authentic

# SPF

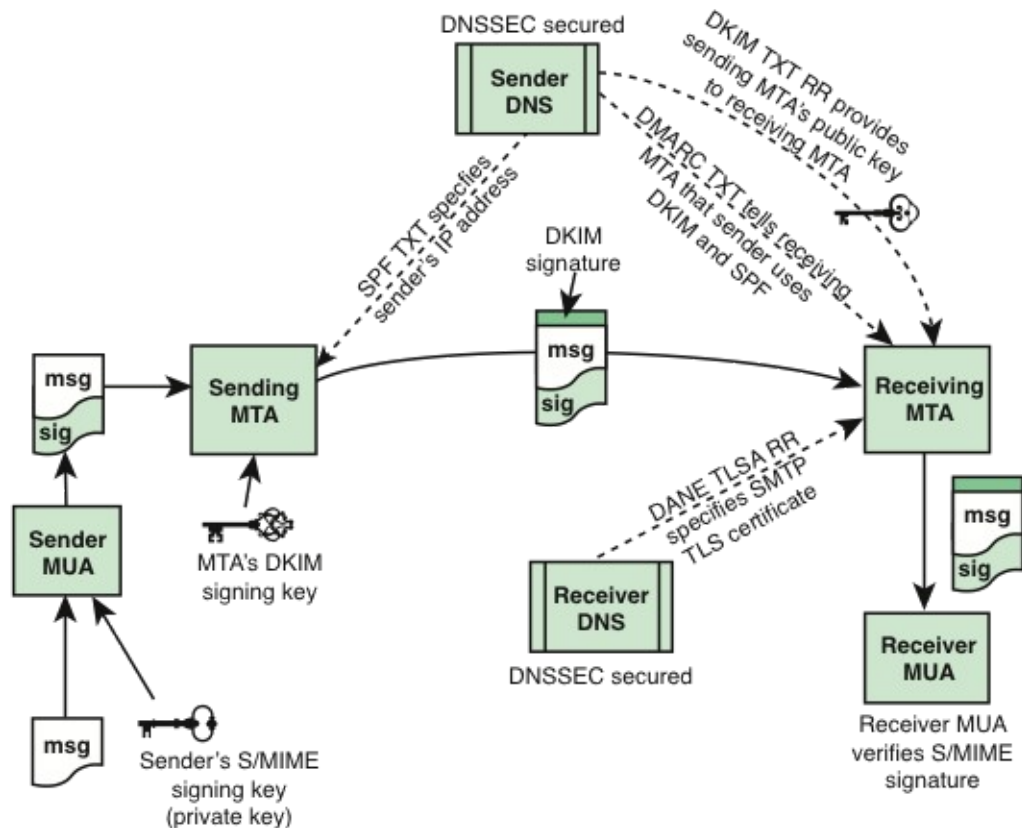
- SPF prevents forgery by designating a handful of IP addresses where emails can originate for your domain. SPF check will fail if a spammer sends a forged message from their own IP, allowing Xeams and other servers to treat that as spam.

It is very easy to forge a sender's email address but is very difficult to forge an IP address. This makes SPF very effective in combating email forgeries.

- Prevents email forgery by confirming an incoming message came from an IP address designated by the sender. SPF checks the **MAIL FROM** value in the SMTP Envelope conversation. It does not check the **FROM** header in the actual message.







DANE = DNS-based Authentication of Named Entities  
 DKIM = DomainKeys Identified Mail  
 DMARC = Domain-based Message Authentication, Reporting, and Conformance  
 DNSSEC = Domain Name System Security Extensions  
 SPF = Sender Policy Framework  
 S/MIME = Secure Multi-Purpose Internet Mail Extensions  
 TLSA RR = Transport Layer Security Authentication Resource Record

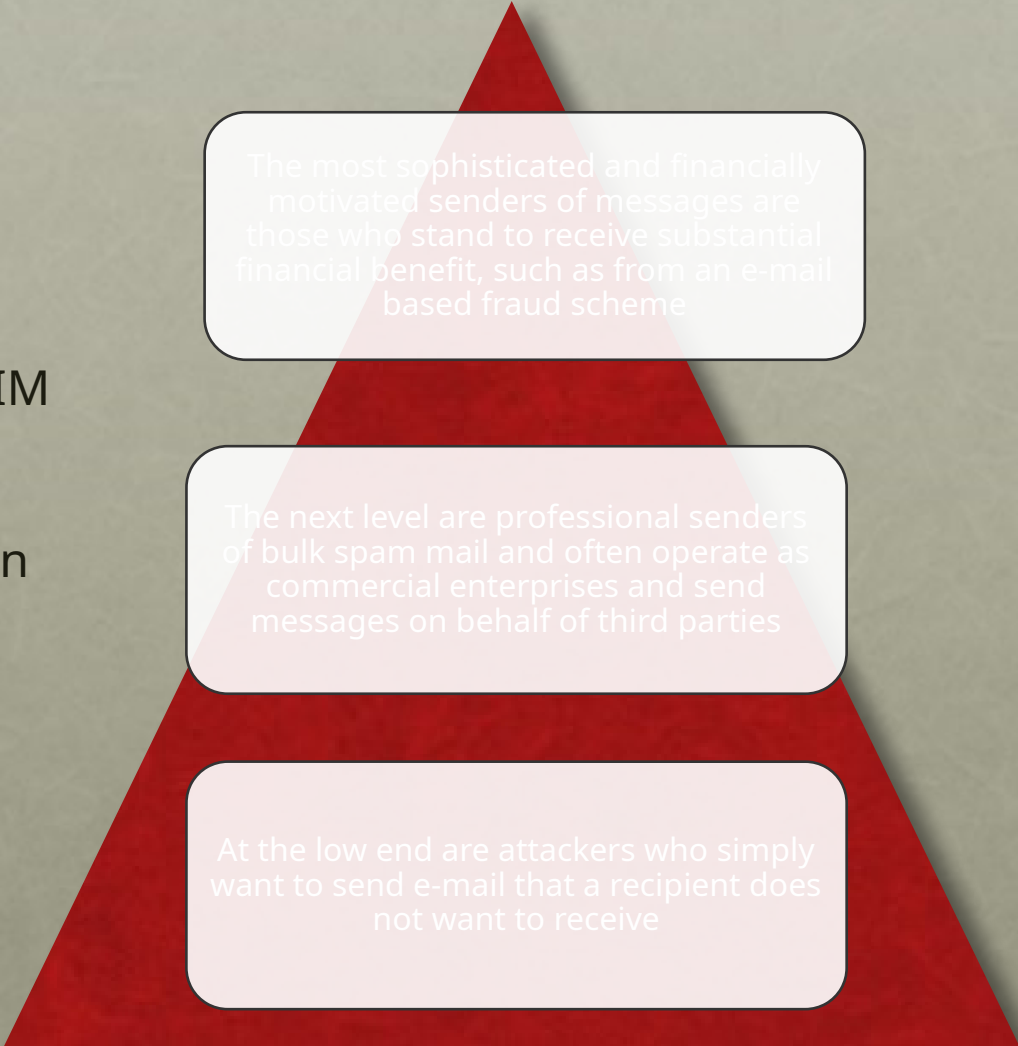
**Figure 8.4 The Interrelationship of DNSSEC, SPF, DKIM, DMARC, DANE, and S/MIME for Assuring Message Authenticity and Integrity**

# 3) DomainKeys Identified Mail (DKIM)

- A specification for cryptographically signing e-mail messages, permitting a signing domain to claim responsibility for a message in the mail stream
- Message recipients can verify the signature by querying the signer's domain directly to retrieve the appropriate public key and can thereby confirm that the message was attested to by a party in possession of the private key for the signing domain
- Proposed Internet Standard RFC 6376
- Has been widely adopted by a range of e-mail providers and Internet Service Providers (ISPs)

# E-mail Threats

- RFC 4686 (*Analysis of Threats Motivating DomainKeys Identified Mail*)
  - Describes the threats being addressed by DKIM in terms of the characteristics, capabilities, and location of potential attackers
- Characterized on three levels of threat:



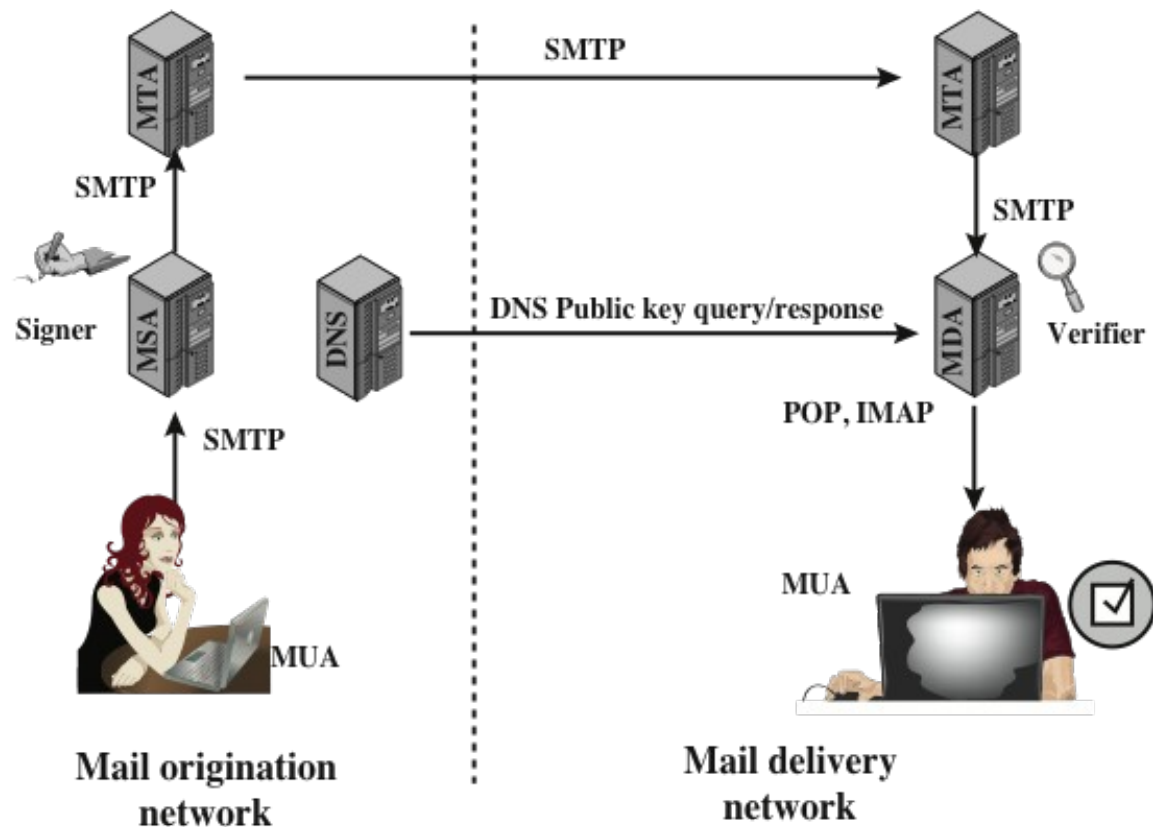
The most sophisticated and financially motivated senders of messages are those who stand to receive substantial financial benefit, such as from an e-mail based fraud scheme

The next level are professional senders of bulk spam mail and often operate as commercial enterprises and send messages on behalf of third parties

At the low end are attackers who simply want to send e-mail that a recipient does not want to receive

# Dkim strategy

- DKIM is designed to provide an e-mail authentication technique that is transparent to the end user
- In essence, a user's e-mail message is signed by a private key of the administrative domain from which the e-mail originates
- The signature covers all of the content of the message and some of the RFC 5322 message headers
- This approach differs from that of S/MIME, which use the originator's private key to sign the content of the message
- The motivation for DKIM is based on the following reasoning: DKIM is not implemented in client programs (MUAs) and is therefore transparent to the user; the user need not take any action
  - DKIM applies to all mail from cooperating domains
  - DKIM allows good senders to prove that they did send a particular message and to prevent forgers from masquerading as good senders



DNS = domain name system  
MDA = mail delivery agent  
MSA = mail submission agent  
MTA = message transfer agent  
MUA = message user agent

**Figure 8.10 Simple Example of DKIM Deployment**



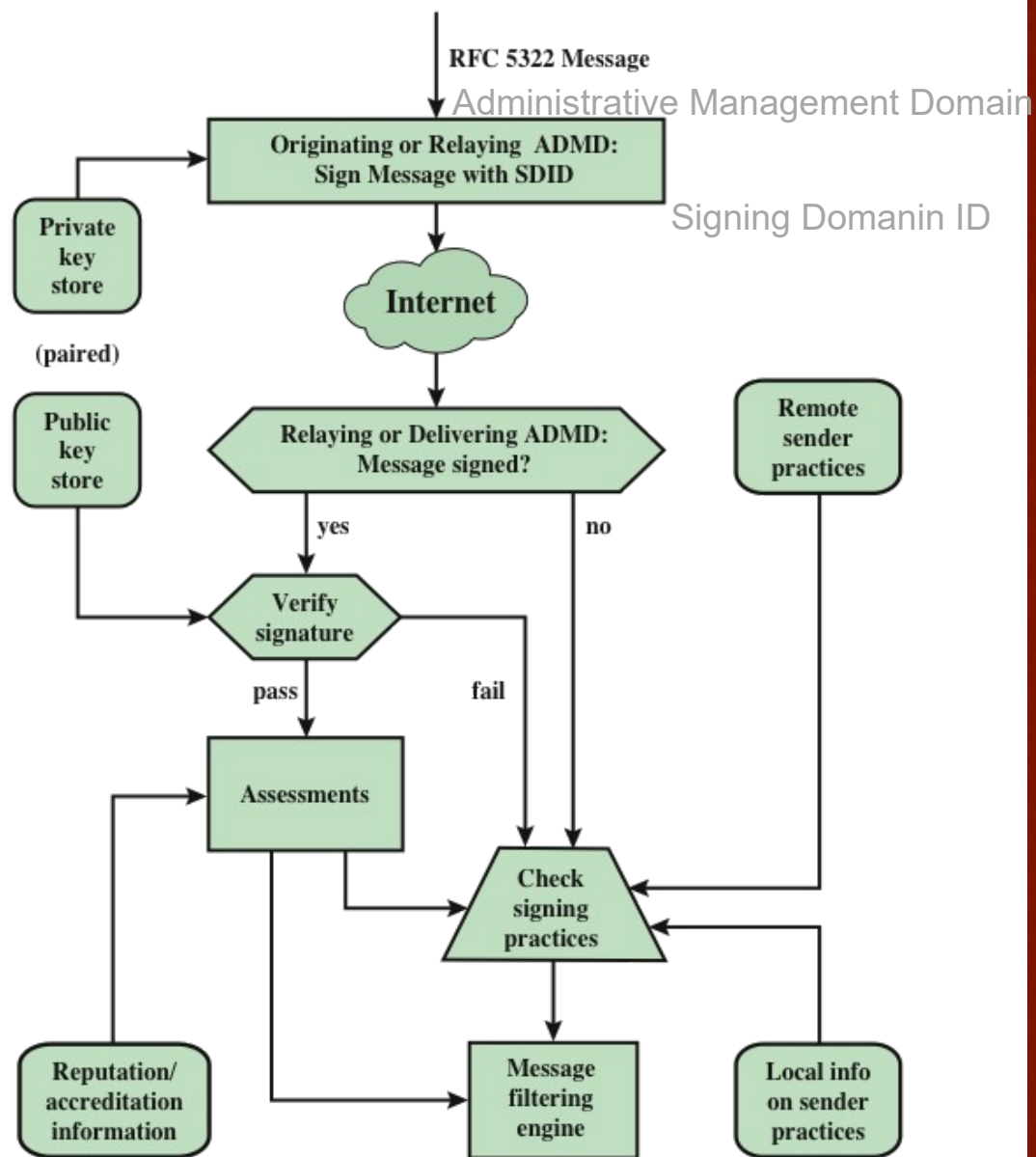
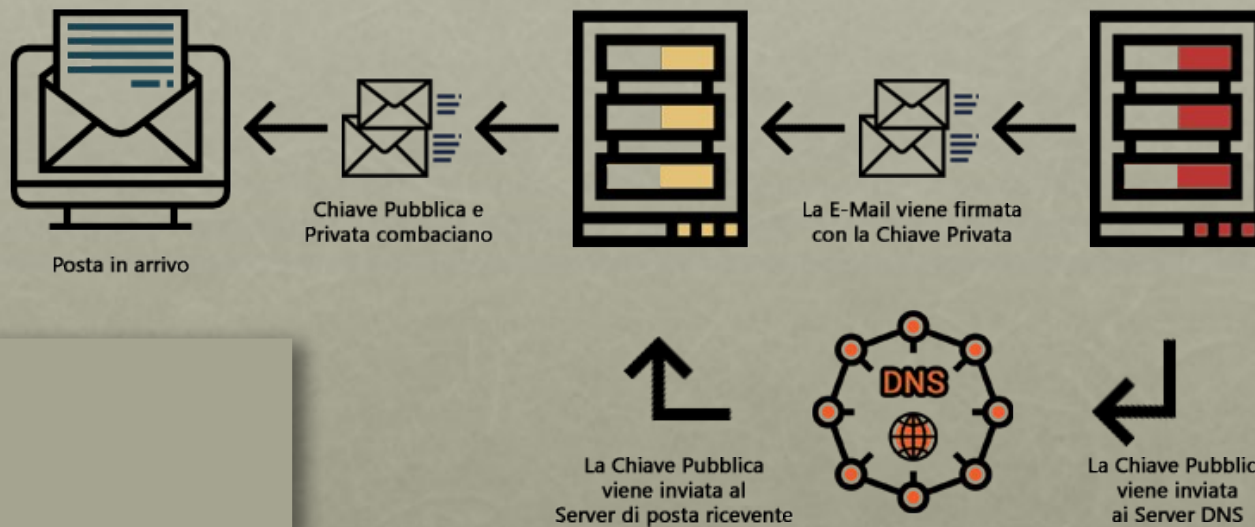


Figure 8.11 DKIM Functional Flow

# DKIM

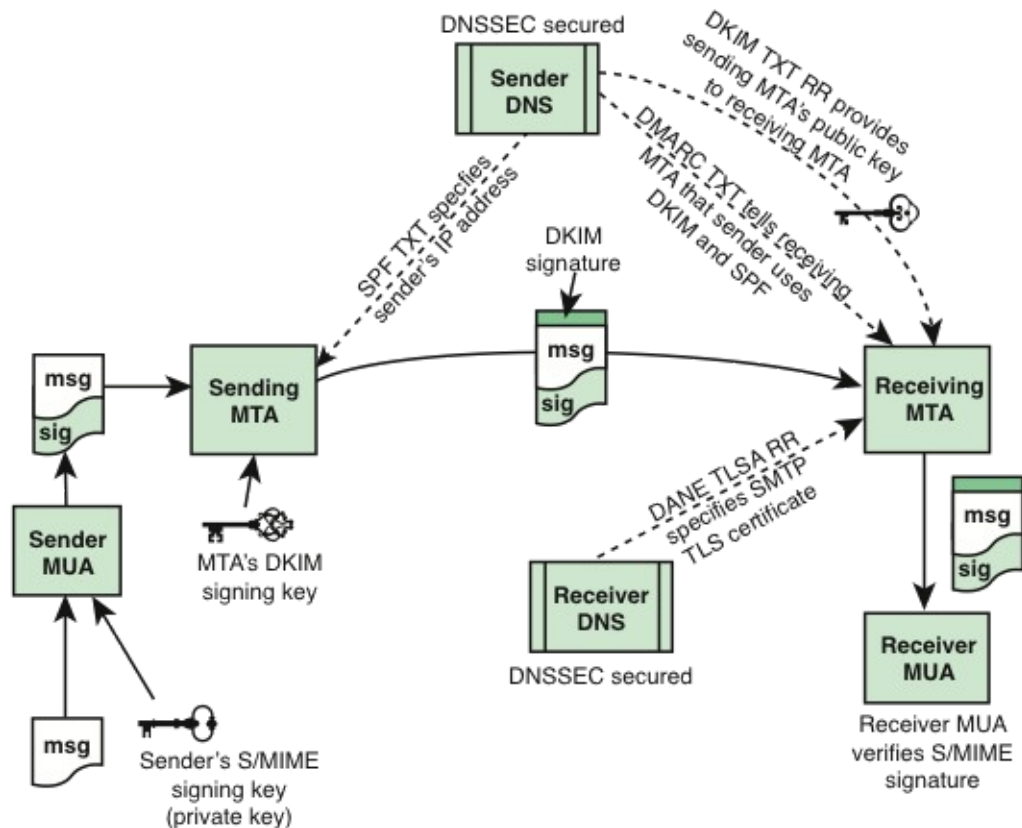
- DKIM ensures the message was actually composed by someone belonging to the domain they claim to be part of. Additionally, it ensures the message is not modified in transit. Technically, the message is digitally signed by the email server before leaving and this signature is verified by the receiving end, allowing the receiving end to verify the authenticity of the message.

Confirms the content of the message was not modified during transit and the message originated from the sender's domain. This protocol emphasizes on email's domain name rather than the IP address where message came from.



# Dkim lookup

- Cerca selector nella mail ricevuta da un dominio
- <https://dmarcian.com/dkim-inspector/>



DANE = DNS-based Authentication of Named Entities  
 DKIM = DomainKeys Identified Mail  
 DMARC = Domain-based Message Authentication, Reporting, and Conformance  
 DNSSEC = Domain Name System Security Extensions  
 SPF = Sender Policy Framework  
 S/MIME = Secure Multi-Purpose Internet Mail Extensions  
 TLSA RR = Transport Layer Security Authentication Resource Record

**Figure 8.4 The Interrelationship of DNSSEC, SPF, DKIM, DMARC, DANE, and S/MIME for Assuring Message Authenticity and Integrity**



## 4) Domain-based message authentication, reporting, and conformance (dmarc)

- DMARC allows e-mail 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
- Is defined in RFC 7489, *Domain-based Message Authentication, Reporting, and Conformance*, March 2015
- Works with SPF and DKIM
- DMARC standardizes how e-mail receivers perform e-mail authentication using SPF and DKIM mechanisms
- DMARC authentication deals with the From domain in the message header, as defined in RFC 5322
- DMARC requires that From address match (be aligned with) an Authenticated Identifier from DKIM or SPF

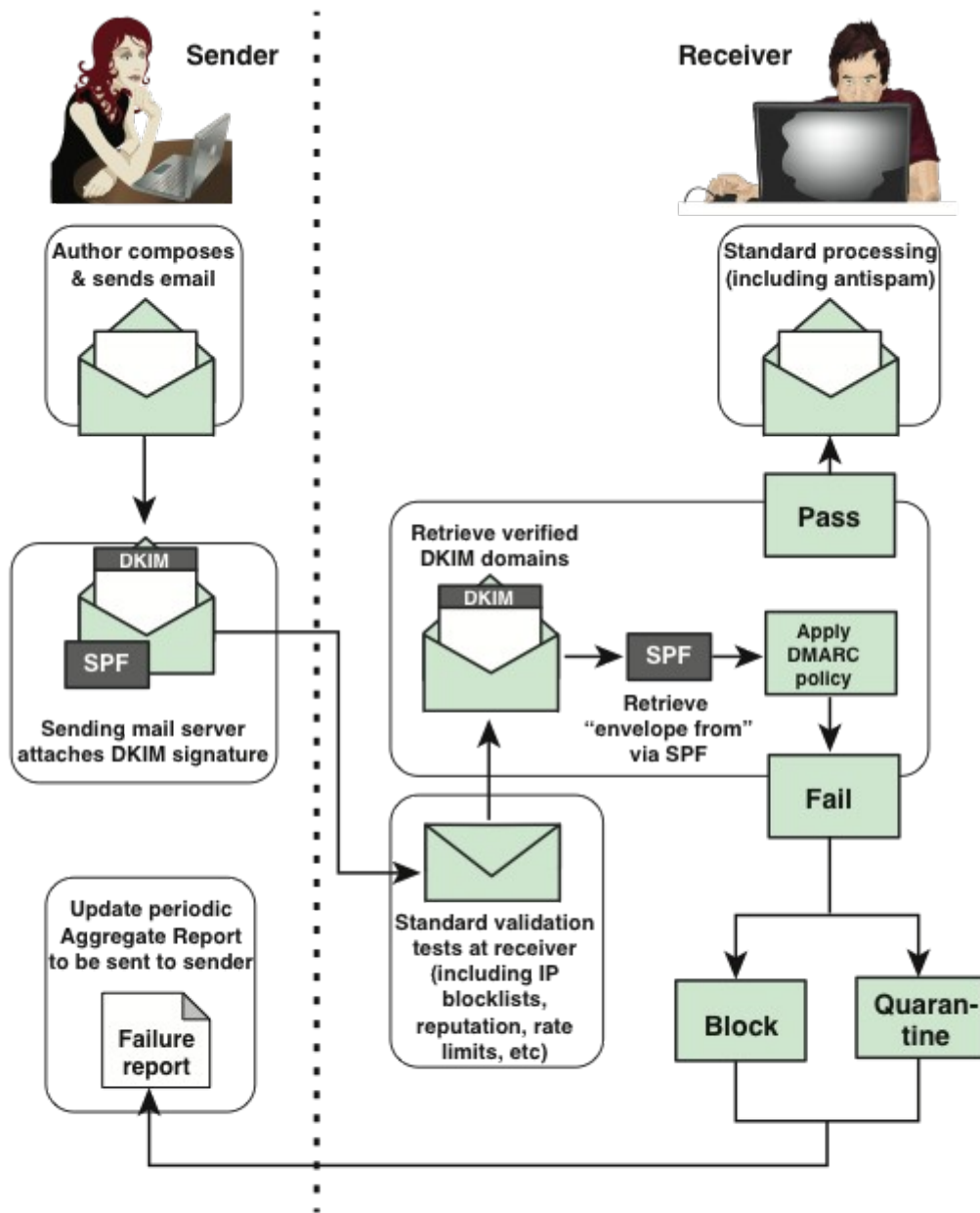


Figure 8.12 DMARC Functional Flow

# Table 8.8

## DMARC Tag and Value Descriptions

(Table can be found on page 281 in the textbook)

Tag (Name)	Description
v= (Version)	Version field that must be present as the first element. By default the value is always <b>DMARC1</b> .
p= (Policy)	Mandatory policy field. May take values <b>none</b> or <b>quarantine</b> or <b>reject</b> . This allows for a gradually tightening policy where the sender domain recommends no specific action on mail that fails DMARC checks ( <b>p=none</b> ), through treating failed mail as suspicious ( <b>p=quarantine</b> ), to rejecting all failed mail ( <b>p=reject</b> ), preferably at the SMTP transaction stage.
aspf= (SPF Policy)	Values are <b>r</b> (default) for relaxed and <b>s</b> for strict SPF domain enforcement. Strict alignment requires an exact match between the From address domain and the (passing) SPF check must exactly match the MailFrom address (HELO address). Relaxed requires that only the From and MailFrom address domains be in alignment. For example, the MailFrom address domain <b>smtp.example.org</b> and the From address <b>announce@example.org</b> are in alignment, but not a strict match.
adkim = (DKIM Policy)	Optional. Values are <b>r</b> (default) for relaxed and <b>s</b> for strict DKIM domain enforcement. Strict alignment requires an exact match between the From domain in the message header and the DKIM domain presented in the <b>d=</b> DKIM tag. Relaxed requires only that the domain part is in alignment (as in <b>aspf</b> ).
fo= (Failure reporting options)	Optional. Ignore if a <b>ruf</b> argument is not also present. Value <b>0</b> indicates the receiver should generate a DMARC failure report if all underlying mechanisms fail to produce an aligned pass result. Value <b>1</b> means generate a DMARC failure report if any underlying mechanism produces something other than an aligned pass result. Other possible values are <b>d</b> (generate a DKIM failure report if a signature failed evaluation), and <b>s</b> (generate an SPF failure report if the message failed SPF evaluation). These values are not exclusive and may be combined.
ruf=	Optional, but requires the <b>fo</b> argument to be present. Lists a series of URIs (currently just <b>mailto:&lt;emailaddress&gt;</b> ) that list where to send forensic feedback reports. This is for reports on message specific failures.
rua=	Optional list of URIs (like in <b>ruf=</b> , using the <b>mailto:</b> URI) listing where to send aggregate feedback back to the sender. These reports are sent based on the interval requested using the <b>ri=</b> option, with a default of 86400 seconds if not listed.
ri= (Reporting interval)	Optional with the default value of 86400 seconds. The value listed is the reporting interval desired by the sender.
pct= (Percent)	Optional with the default value of <b>100</b> . Expresses the percentage of a sender's mail that should be subject to the given DMARC policy. This allows senders to ramp up their policy enforcement gradually and prevent having to commit to a rigorous policy before getting feedback on their existing policy.
sp= (Receiver policy)	Optional with a default value of <b>none</b> . Other values include the same range of values as the <b>p=</b> argument. This is the policy to be applied to mail from all identified subdomains of the given DMARC RR.

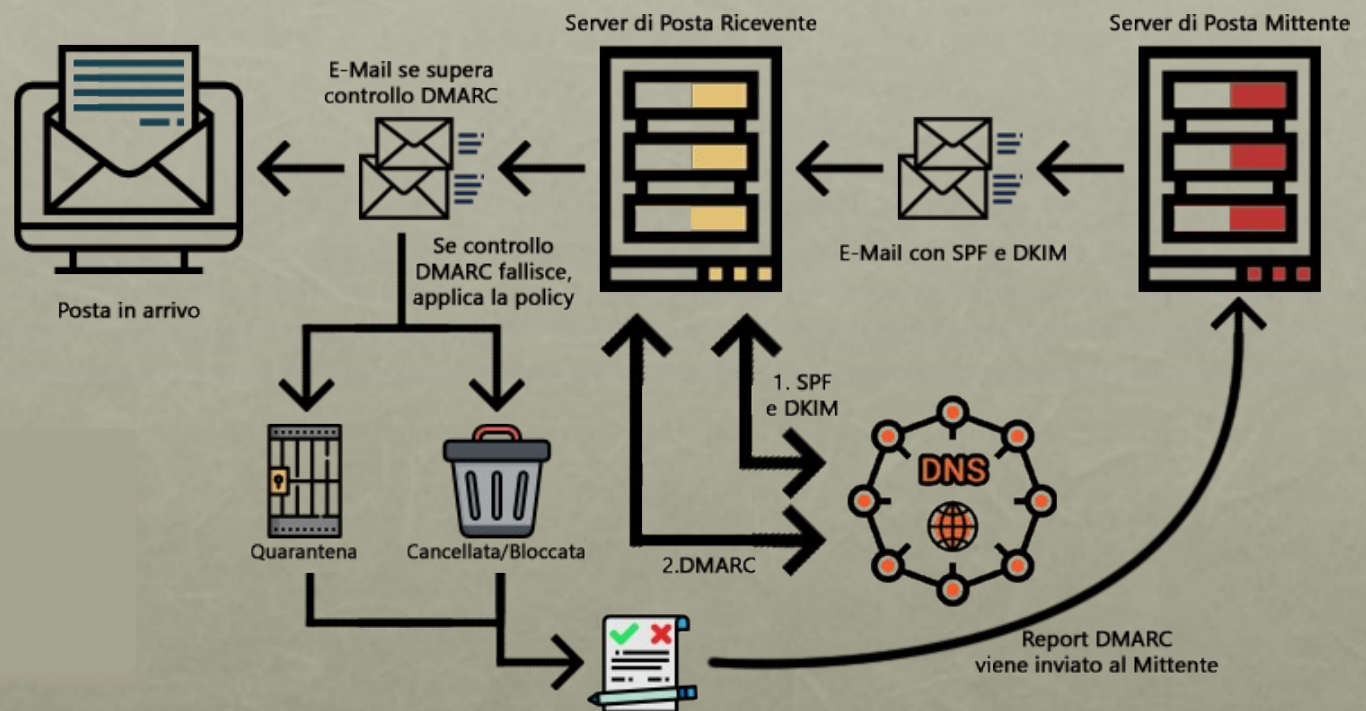
# DMARC

- DMARC builds on top of SPF and DKIM. This not only helps prevent forgery but also provides a sophisticated reporting mechanism allowing the senders to fine tune their SPF and DKIM rules.

Unlike SPF, DMARC looks at the **FROM** header of an email. An incoming email is considered to be "DMARC Aligned" if the domain name of **FROM** header matches with the domain name of the **MAIL FROM** value in the envelope. Additionally, it also checks if the domain specified in the **FROM** header matches with the domain name specified in the DKIM signature.

Aside from checking for message alignment, which prevents forgery, DMARC also provides a mechanism for email servers to report their discovery to other servers on the internet. For example, servers for gmail.com and yahoo.com will send reports once a day to your Xeams explaining how they treated messages that came from your domain.





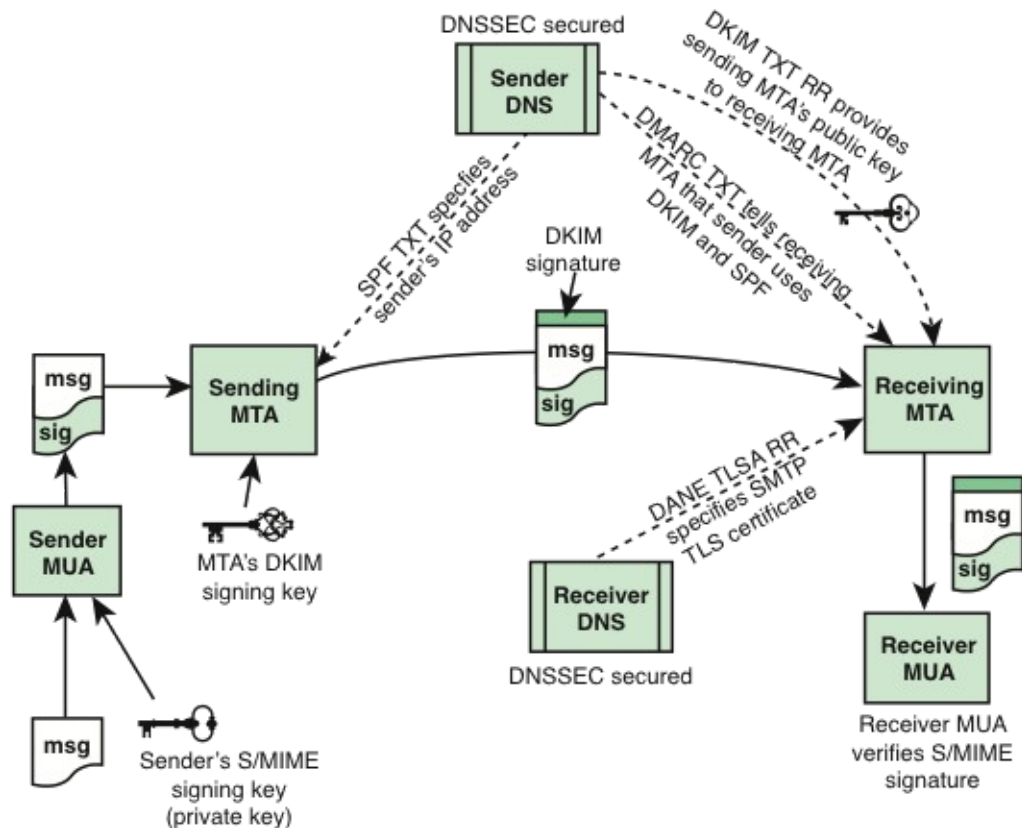


# Dmark check

- <https://mxtoolbox.com/dmarc.aspx>

# Summary

- Internet mail architecture
- E-mail formats
- E-mail threats and comprehensive e-mail security
- Pretty good privacy
- DomainKeys Identified Mail
- S/MIME
- DNSSEC
- DNS-based authentication of named entities
- Sender policy framework
- Domain-based message authentication, reporting, and conformance



DANE = DNS-based Authentication of Named Entities  
 DKIM = DomainKeys Identified Mail  
 DMARC = Domain-based Message Authentication, Reporting, and Conformance  
 DNSSEC = Domain Name System Security Extensions  
 SPF = Sender Policy Framework  
 S/MIME = Secure Multi-Purpose Internet Mail Extensions  
 TLSA RR = Transport Layer Security Authentication Resource Record

**Figure 8.4 The Interrelationship of DNSSEC, SPF, DKIM, DMARC, DANE, and S/MIME for Assuring Message Authenticity and Integrity**

- <https://www.bind9.net/dns-tools>
- <https://zonemaster.iis.se/en/>
- <https://dnssec-analyzer.verisignlabs.com/>
- Free s/mime certificates
  - <https://extrassl.actalis.it/portal/uapub/freemail?lang=en>

# CONCLUSIONI

- Le estensioni per la sicurezza provvedono alla:
  - **Protezione dei trasferimenti su Internet:**
    - Dati firmati con chiavi pubbliche (**SIG**, **KEY**).
    - L'assenza di dati DNS è notificata (**NXT**).
  - **Protezione per i trasferimenti DNS locali:**
    - I messaggi tra NS e resolver sono autenticati (**TSIG**).
    - Trasferimenti di zona tra NS primari e secondari (**TSIG**).
  - **Infrastruttura a chiave pubblica PKI:**
    - Distribuzione di chiavi pubbliche per altri protocolli di sicurezza (**KEY**).
    - Distribuzione di diversi tipi di certificati (**CERT**).





# Autenticità ed Integrità dei Dati

- Autenticità: I dati sono pubblicati dall'entità che si ritiene ne abbia l'autorità???
- Integrità: I dati ricevuti sono i medesimi che sono stati pubblicati???
- La crittografia a chiave asimmetrica risponde alle seguenti esigenze:
  - Verificare l'integrità e l'autenticità dei dati attraverso le signatures
  - Verificare l'autenticità delle signatures
- L'integrità e l'autenticità dei dati ottenuta firmando i RRs con la chiave privata
- La chiave pubblica DNSKEY usata per verificare i RRSIGs
- I nodi '*figlio*' firmano le proprie zone con la propria chiave privata
- L'autenticità della chiave è stabilita dal *parent* attraverso la coppia *signature/checksum*

# Resource Records

- 3 RRs correlati alla cifratura a chiave asimmetrica
  - RRSIG -> signature su RRSets ottenuta usando la chiave privata
  - DNSKEY -> Chiave pubblica, necessaria per verificare un RRSIG
  - DS -> Delegation Signer: puntatore per la costruzione di catene di autenticazione
- 1 RR per la consistenza interna
  - NSEC -> Indica quale nome sia il successivo nella zona

# RRs e RRSets

- Resource Record

– Name	TTL	class	type	rdata
– <i>www.example.net.</i>	<i>7200</i>	<i>IN</i>	<i>A</i>	<i>192.168.10.3</i>

- RRSets: RRs con gli stessi *name*, *class* e *type*

– <i>www.example.net.</i>	<i>7200</i>	<i>IN</i>	<i>A</i>	<i>192.168.10.3</i>
			<i>A</i>	<i>10.0.0.3</i>
			<i>A</i>	<i>172.25.215.2</i>

- *Gli RRsets sono firmati, ma non lo sono i singoli RRs che lo compongono*