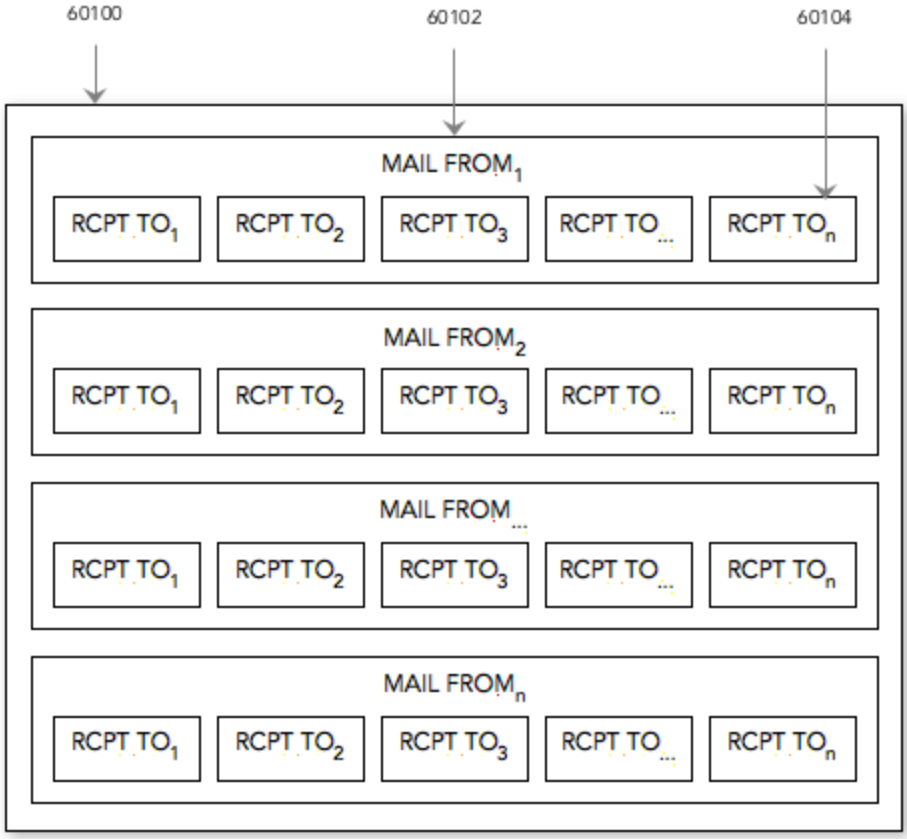

60100
60102
60104
MAIL FROM1
RCPT TO1
RCPT TO2
RCPT TO3
RCPT TO...
RCPT TOn
MAIL FROM2
RCPT TO1
RCPT TO2
RCPT TO3
RCPT TO...
RCPT TOn
MAIL FROM...
RCPT TO1
RCPT TO2
RCPT TO3
RCPT TO...
RCPT TOn
MAIL FROMn
RCPT TO1
RCPT TO2
RCPT TO3
RCPT TO...
RCPT TOn

Fig. 601

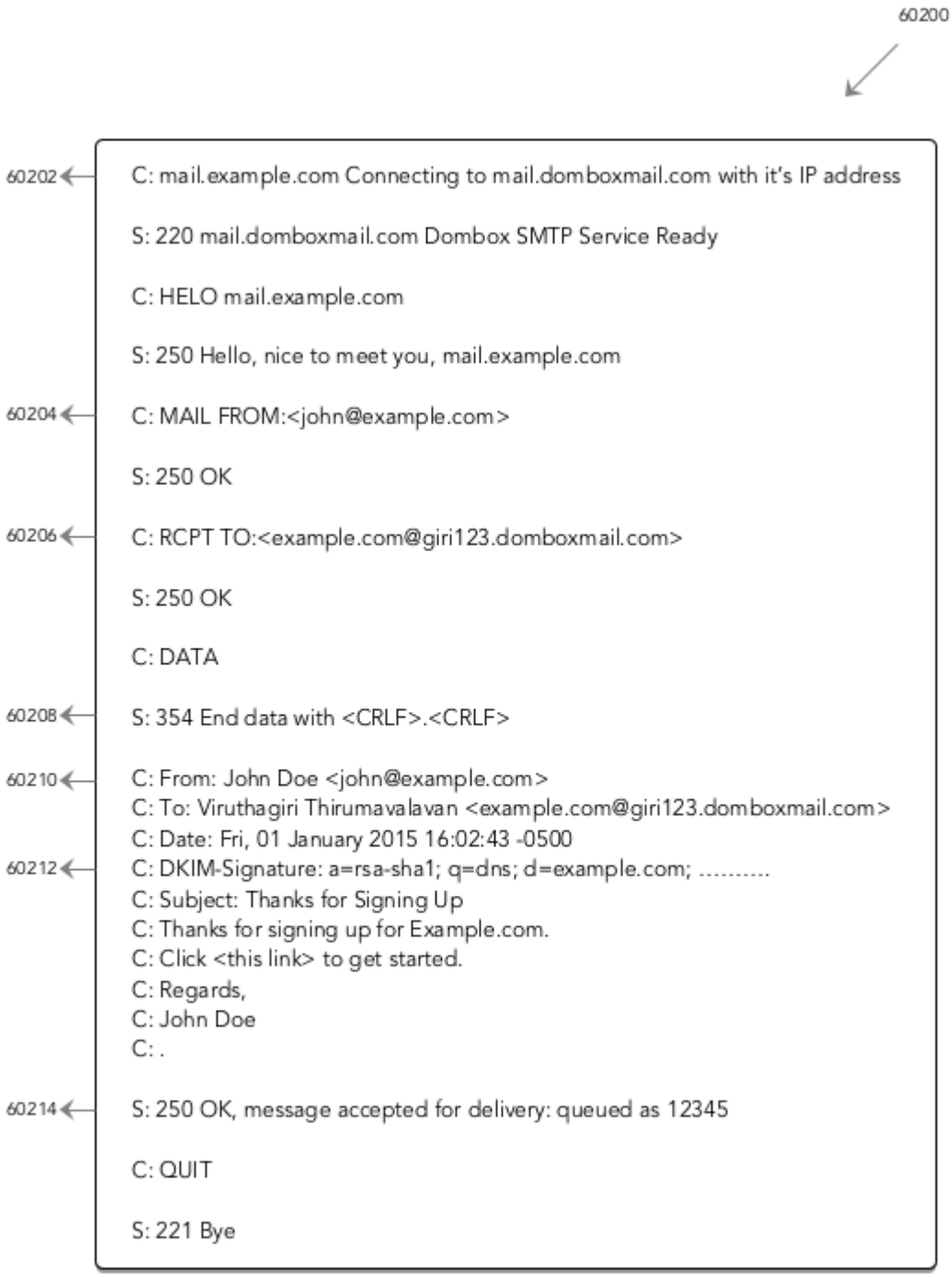

60200
60202
C: mail.example.com Connecting to mail.domboxmail.com with it's IP address
S: 220 mail.domboxmail.com Dombox SMTP Service Ready
C: HELO mail.example.com
S: 250 Hello, nice to meet you, mail.example.com
60204
C: MAIL FROM:<john@example.com>
S: 250 OK
60206
C: RCPT TO:<example.com@giri123.domboxmail.com>
S: 250 OK
C: DATA
60208
S: 354 End data with <CRLF>.<CRLF>
60210
C: From: John Doe <john@example.com>
C: To: Viruthagiri Thirumavalavan <example.com@giri123.domboxmail.com>
C: Date: Fri, 01 January 2015 16:02:43 -0500
60212
C: DKIM-Signature: a=rsa-sha1; q=dns; d=example.com; .....
C: Subject: Thanks for Signing Up
C: Thanks for signing up for Example.com.
C: Click <this link> to get started.
C: Regards,
C: John Doe
C: .
60214
S: 250 OK, message accepted for delivery: queued as 12345
C: QUIT
S: 221 Bye

Fig. 602

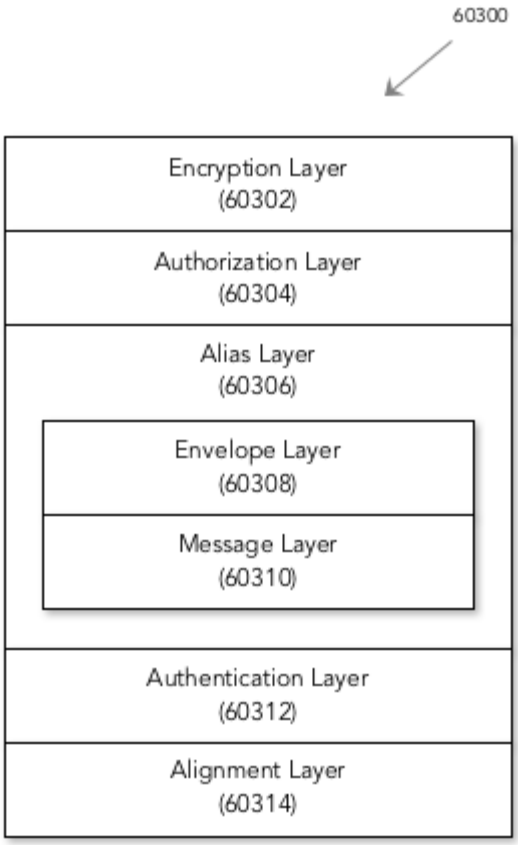

60300
Encryption Layer
(60302)
Authorization Layer
(60304)
Alias Layer
(60306)
Envelope Layer
(60308)
Message Layer
(60310)
Authentication Layer
(60312)
Alignment Layer
(60314)

Fig. 603

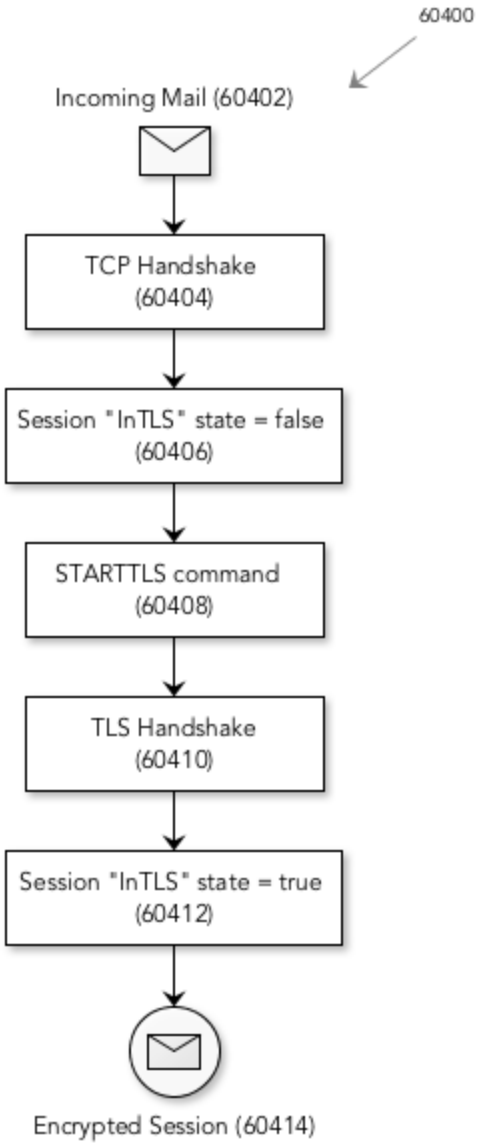

60400
Incoming Mail (60402)
TCP Handshake
(60404)
Session "InTLS" state = false
(60406)
STARTTLS command
(60408)
TLS Handshake
(60410)
Session "InTLS" state = true
(60412)
Encrypted Session (60414)

Fig. 604

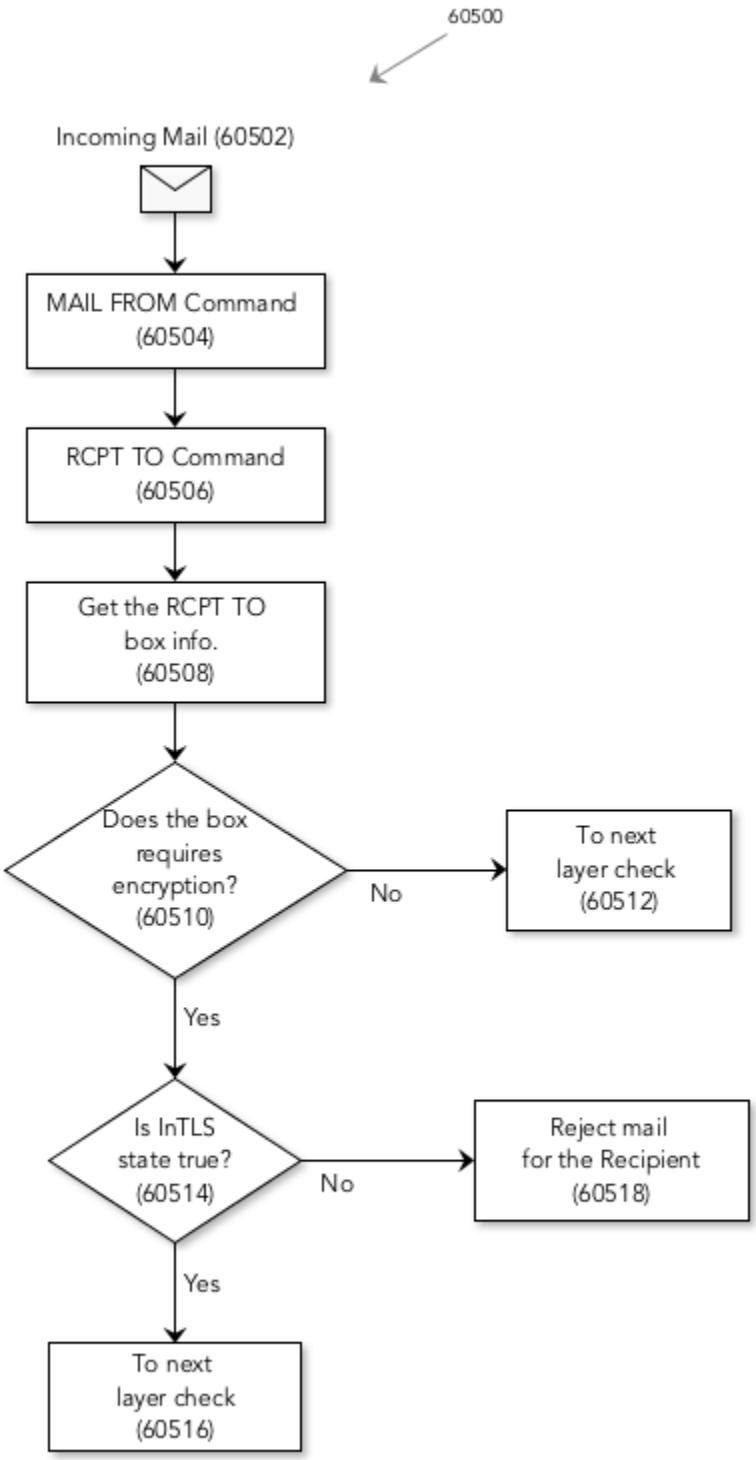

60500
Incoming Mail (60502)
MAIL FROM Command (60504)
RCPT TO Command (60506)
Get the RCPT TO box info. (60508)
Does the box requires encryption? (60510)
No
To next layer check (60512)
Yes
Is InTLS state true? (60514)
No
Reject mail for the Recipient (60518)
Yes
To next layer check (60516)

Fig. 605

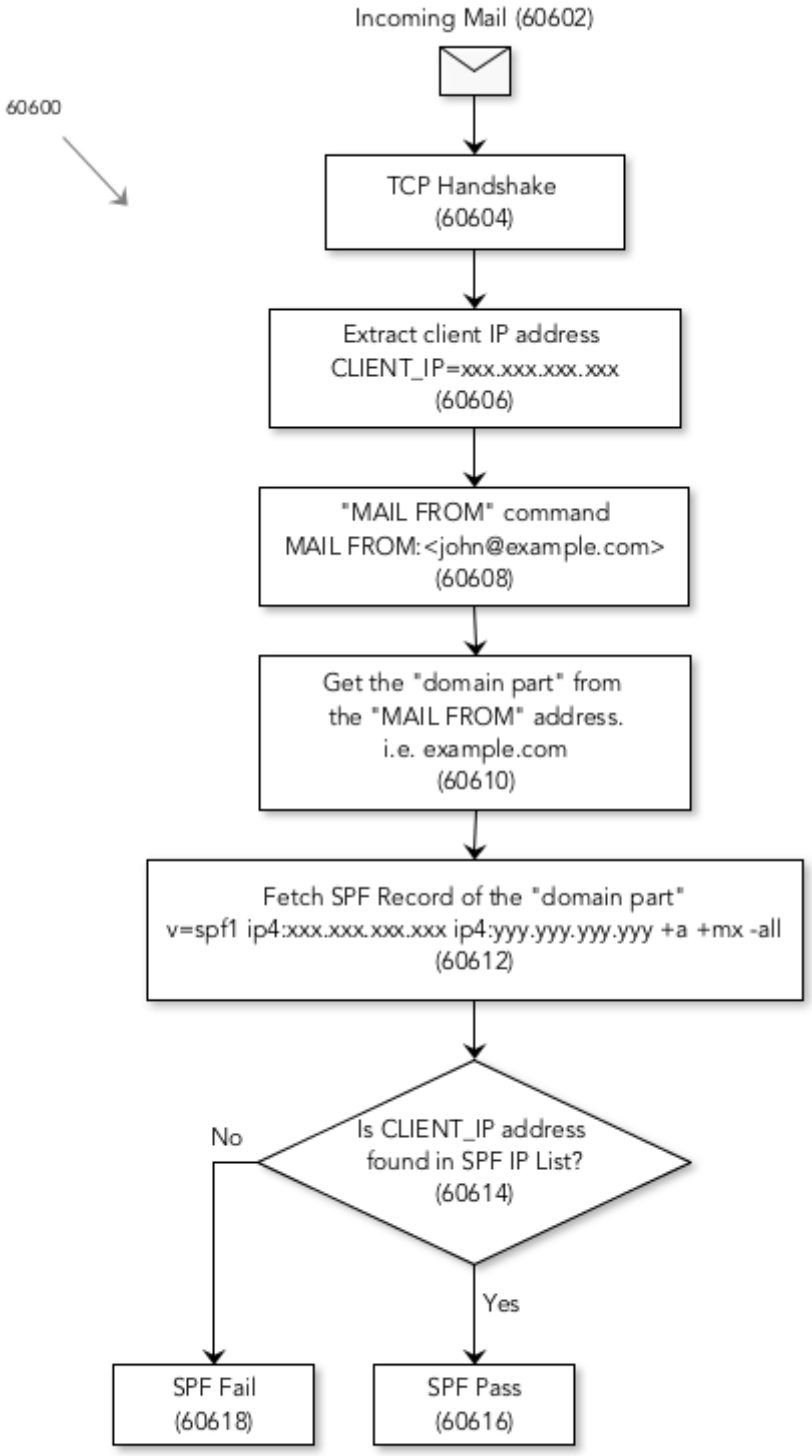

Incoming Mail (60602)
60600
TCP Handshake
(60604)
Extract client IP address
CLIENT_IP=xxx.xxx.xxx.xxx
(60606)
"MAIL FROM" command
MAIL FROM:<john@example.com>
(60608)
Get the "domain part" from
the "MAIL FROM" address.
i.e. example.com
(60610)
Fetch SPF Record of the "domain part"
v=spf1 ip4:xxx.xxx.xxx.xxx ip4:yyy.yyy.yyy.yyy +a +mx -all
(60612)
Is CLIENT_IP address
found in SPF IP List?
(60614)
No
Yes
SPF Fail
(60618)
SPF Pass
(60616)

Fig. 606

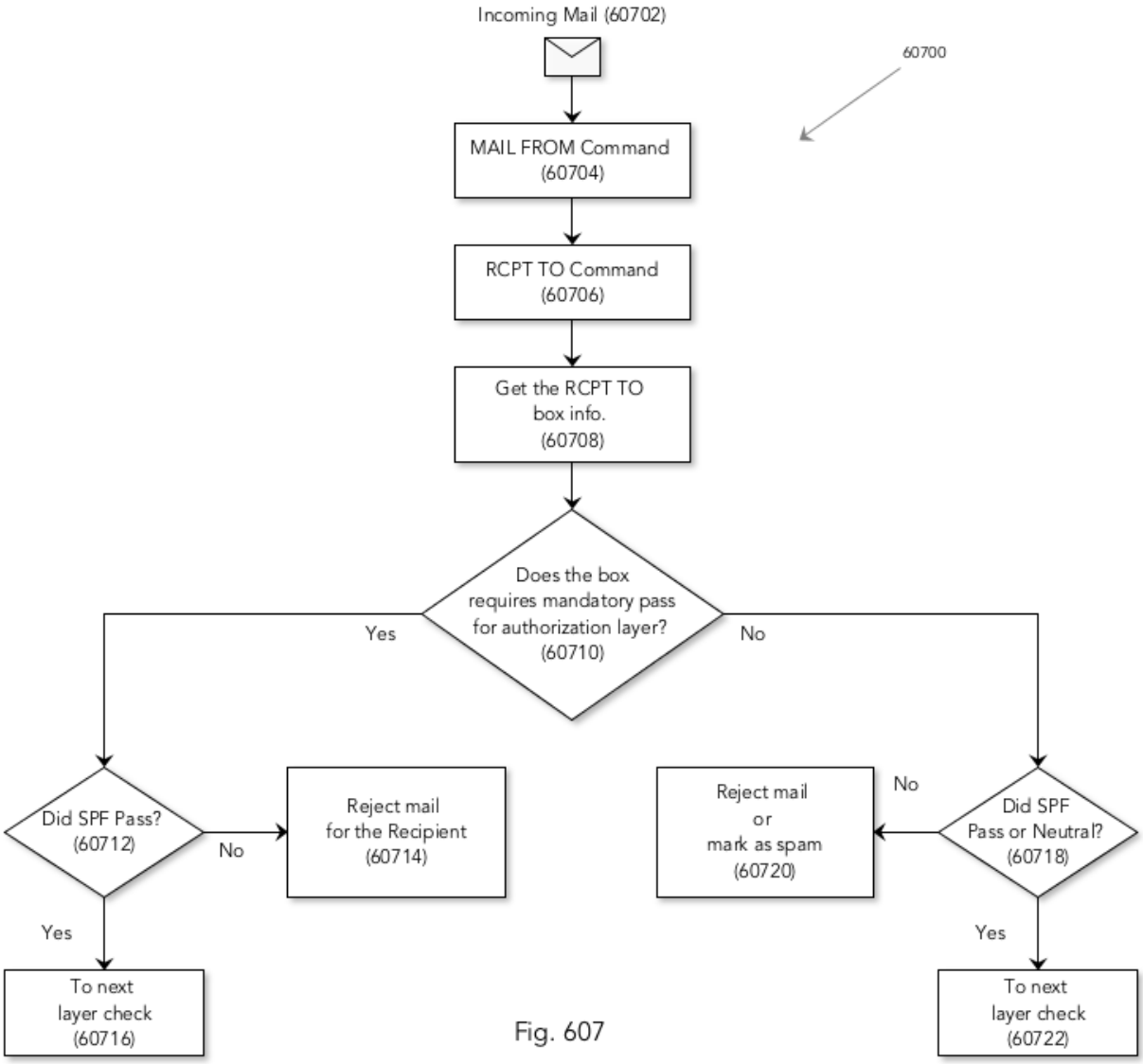

Incoming Mail (60702)
60700
MAIL FROM Command (60704)
RCPT TO Command (60706)
Get the RCPT TO box info. (60708)
Does the box requires mandatory pass for authorization layer? (60710)
Yes
No
Did SPF Pass? (60712)
No
Reject mail for the Recipient (60714)
Yes
To next layer check (60716)
Reject mail or mark as spam (60720)
No
Did SPF Pass or Neutral? (60718)
Yes
To next layer check (60722)

Fig. 607

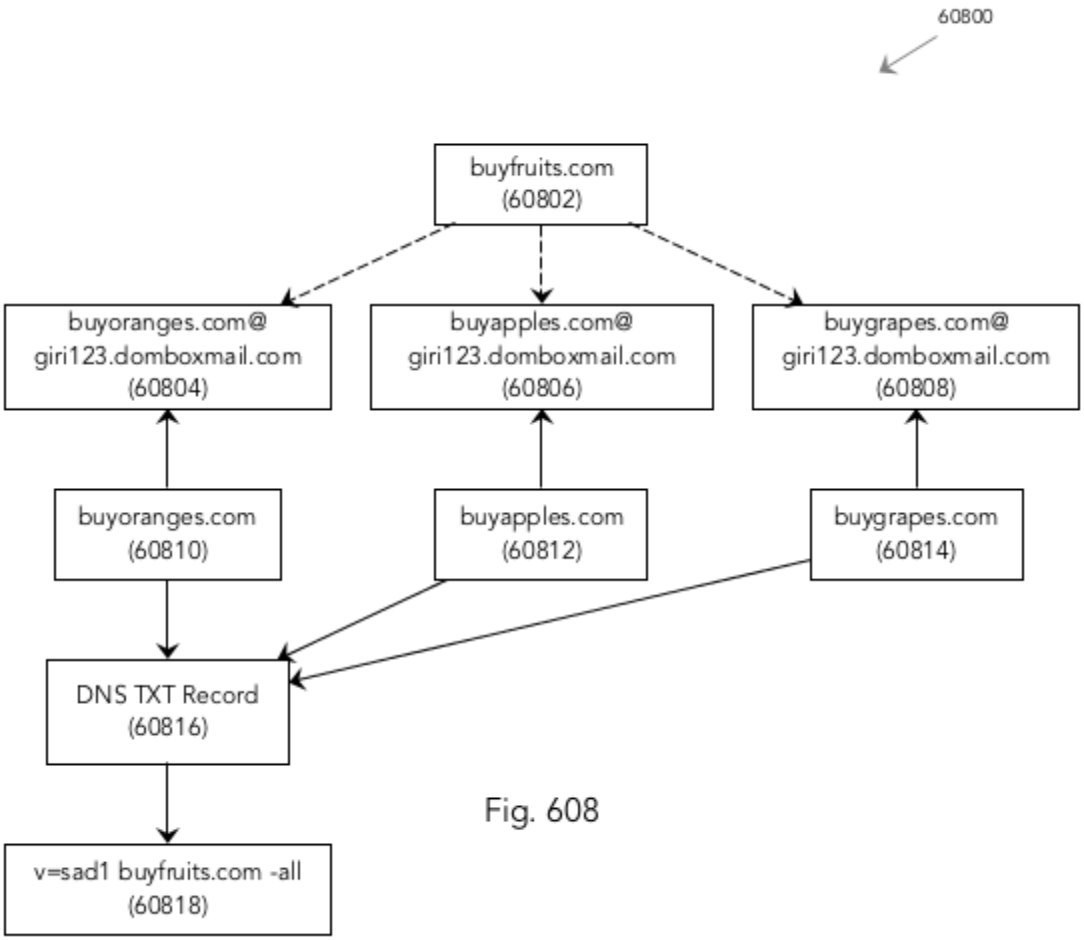

60800
buyfruits.com
(60802)
buyoranges.com@
giri123.domboxmail.com
(60804)
buyapples.com@
giri123.domboxmail.com
(60806)
buygrapes.com@
giri123.domboxmail.com
(60808)
buyoranges.com
(60810)
buyapples.com
(60812)
buygrapes.com
(60814)
DNS TXT Record
(60816)
v=sad1 buyfruits.com -all
(60818)

Fig. 608

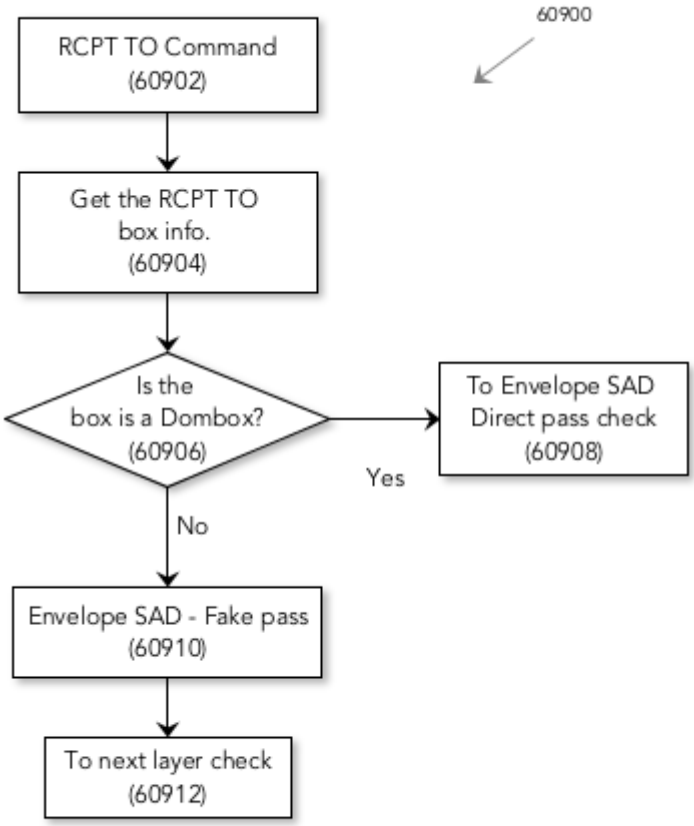

60900
RCPT TO Command
(60902)
Get the RCPT TO
box info.
(60904)
Is the
box is a Dombox?
(60906)
Yes
To Envelope SAD
Direct pass check
(60908)
No
Envelope SAD - Fake pass
(60910)
To next layer check
(60912)

Fig. 609

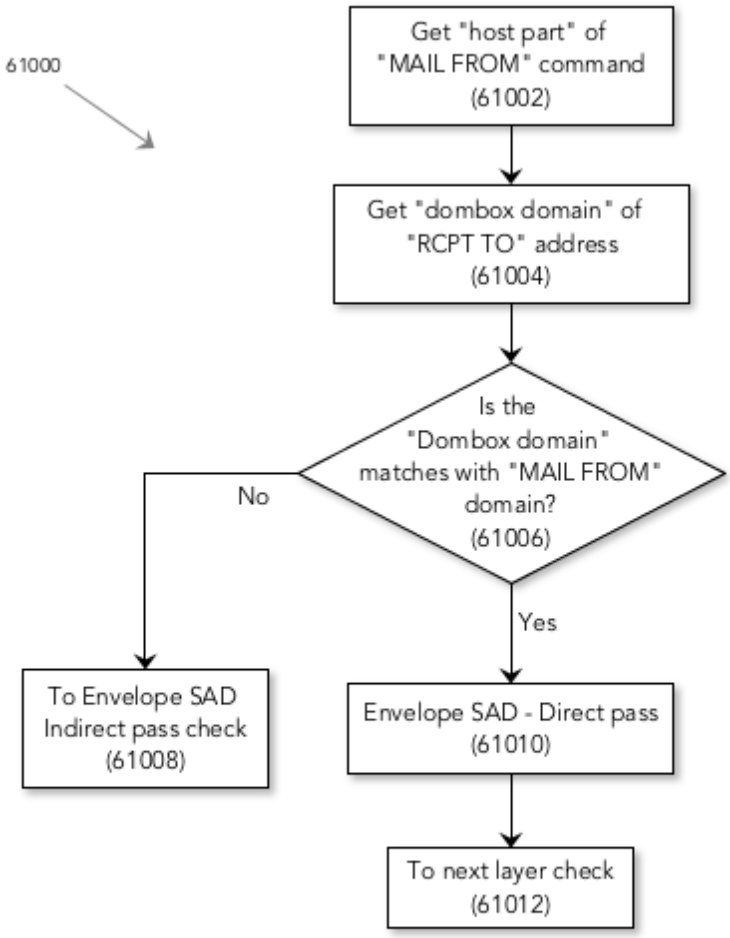
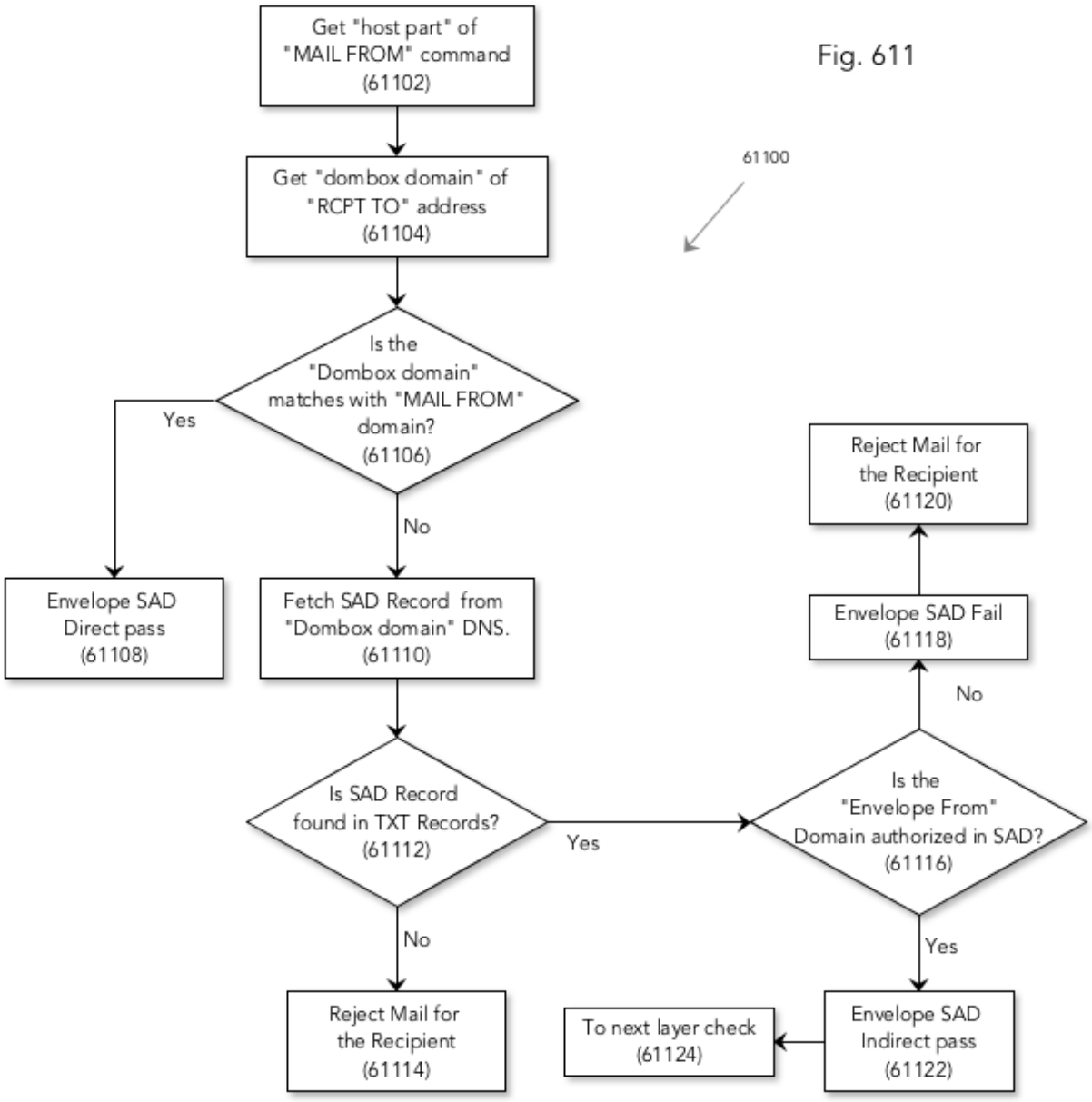

61000
Get "host part" of "MAIL FROM" command (61002)
Get "dombox domain" of "RCPT TO" address (61004)
Is the "Dombox domain" matches with "MAIL FROM" domain? (61006)
No
Yes
To Envelope SAD Indirect pass check (61008)
Envelope SAD - Direct pass (61010)
To next layer check (61012)

Fig. 610

Fig. 611


61100
Get "host part" of "MAIL FROM" command (61102)
Get "dombox domain" of "RCPT TO" address (61104)
Is the "Dombox domain" matches with "MAIL FROM" domain? (61106)
Yes
No
Envelope SAD Direct pass (61108)
Fetch SAD Record from "Dombox domain" DNS. (61110)
Is SAD Record found in TXT Records? (61112)
Yes
No
Reject Mail for the Recipient (61114)
Is the "Envelope From" Domain authorized in SAD? (61116)
No
Yes
Envelope SAD Fail (61118)
Reject Mail for the Recipient (61120)
Envelope SAD Indirect pass (61122)
To next layer check (61124)

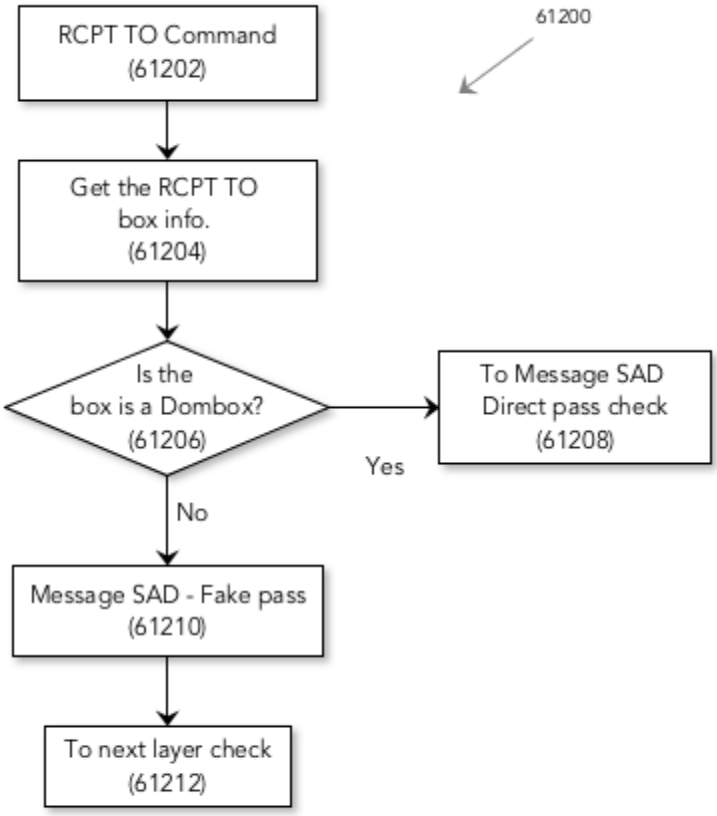

61200
RCPT TO Command
(61202)
Get the RCPT TO
box info.
(61204)
Is the
box is a Dombox?
(61206)
Yes
To Message SAD
Direct pass check
(61208)
No
Message SAD - Fake pass
(61210)
To next layer check
(61212)

Fig. 612

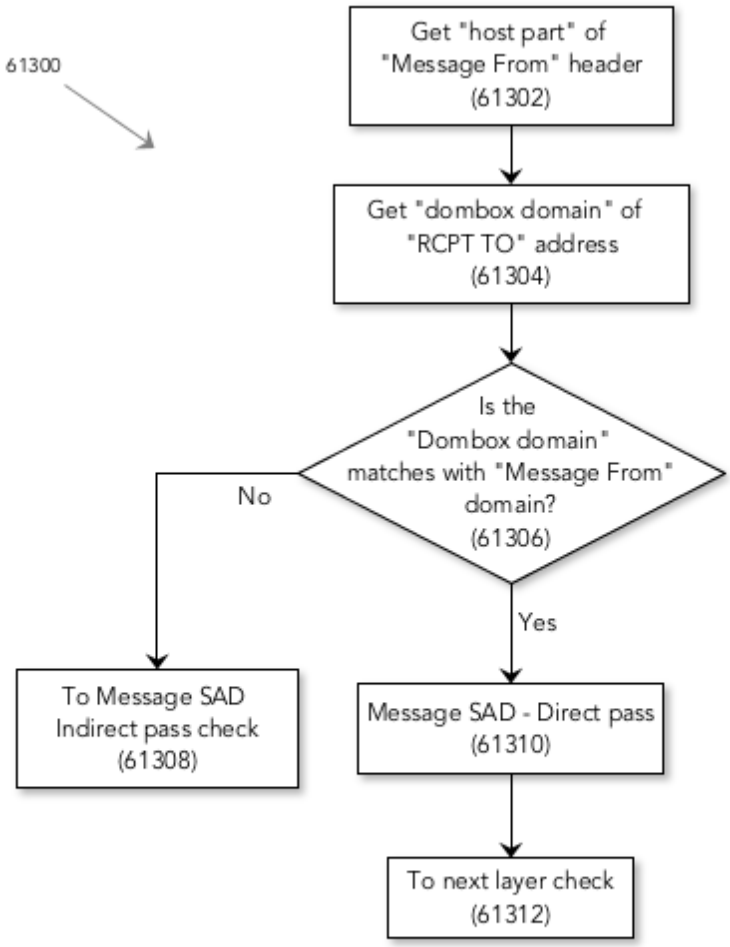
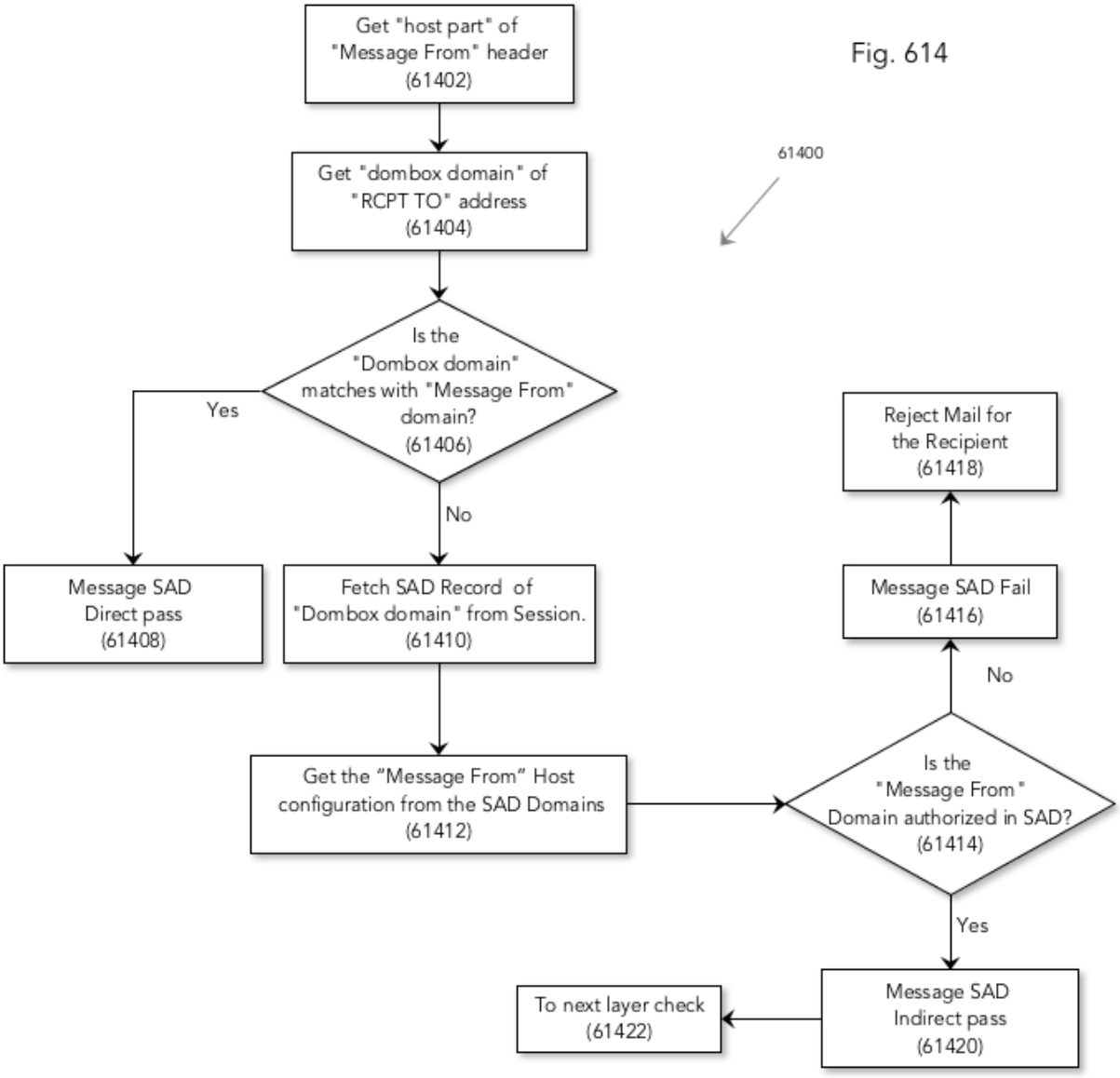

61300
Get "host part" of "Message From" header (61302)
Get "dombox domain" of "RCPT TO" address (61304)
Is the "Dombox domain" matches with "Message From" domain? (61306)
No
Yes
To Message SAD Indirect pass check (61308)
Message SAD - Direct pass (61310)
To next layer check (61312)

Fig. 613

Fig. 614


61400
Get "host part" of "Message From" header (61402)
Get "dombox domain" of "RCPT TO" address (61404)
Is the "Dombox domain" matches with "Message From" domain? (61406)
Yes
No
Message SAD Direct pass (61408)
Fetch SAD Record of "Dombox domain" from Session. (61410)
Get the "Message From" Host configuration from the SAD Domains (61412)
Is the "Message From" Domain authorized in SAD? (61414)
No
Message SAD Fail (61416)
Reject Mail for the Recipient (61418)
Yes
Message SAD Indirect pass (61420)
To next layer check (61422)

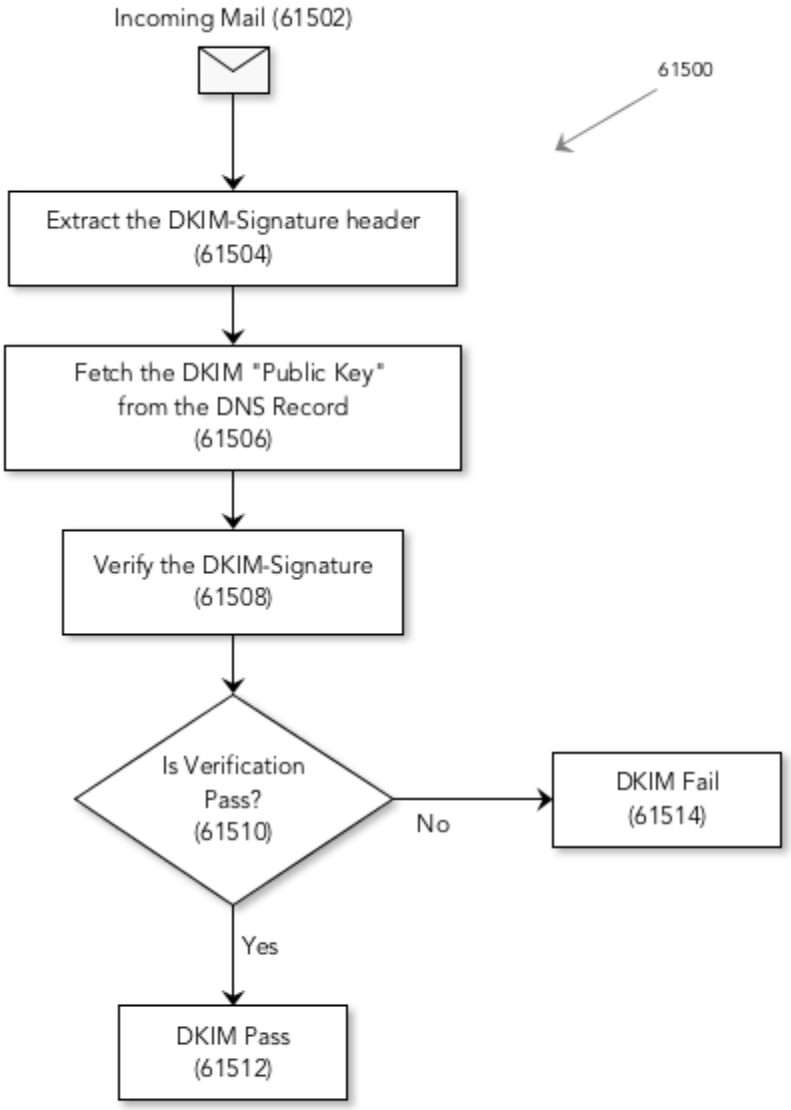

Incoming Mail (61502)
61500
Extract the DKIM-Signature header (61504)
Fetch the DKIM "Public Key" from the DNS Record (61506)
Verify the DKIM-Signature (61508)
Is Verification Pass? (61510)
No
DKIM Fail (61514)
Yes
DKIM Pass (61512)

Fig. 615

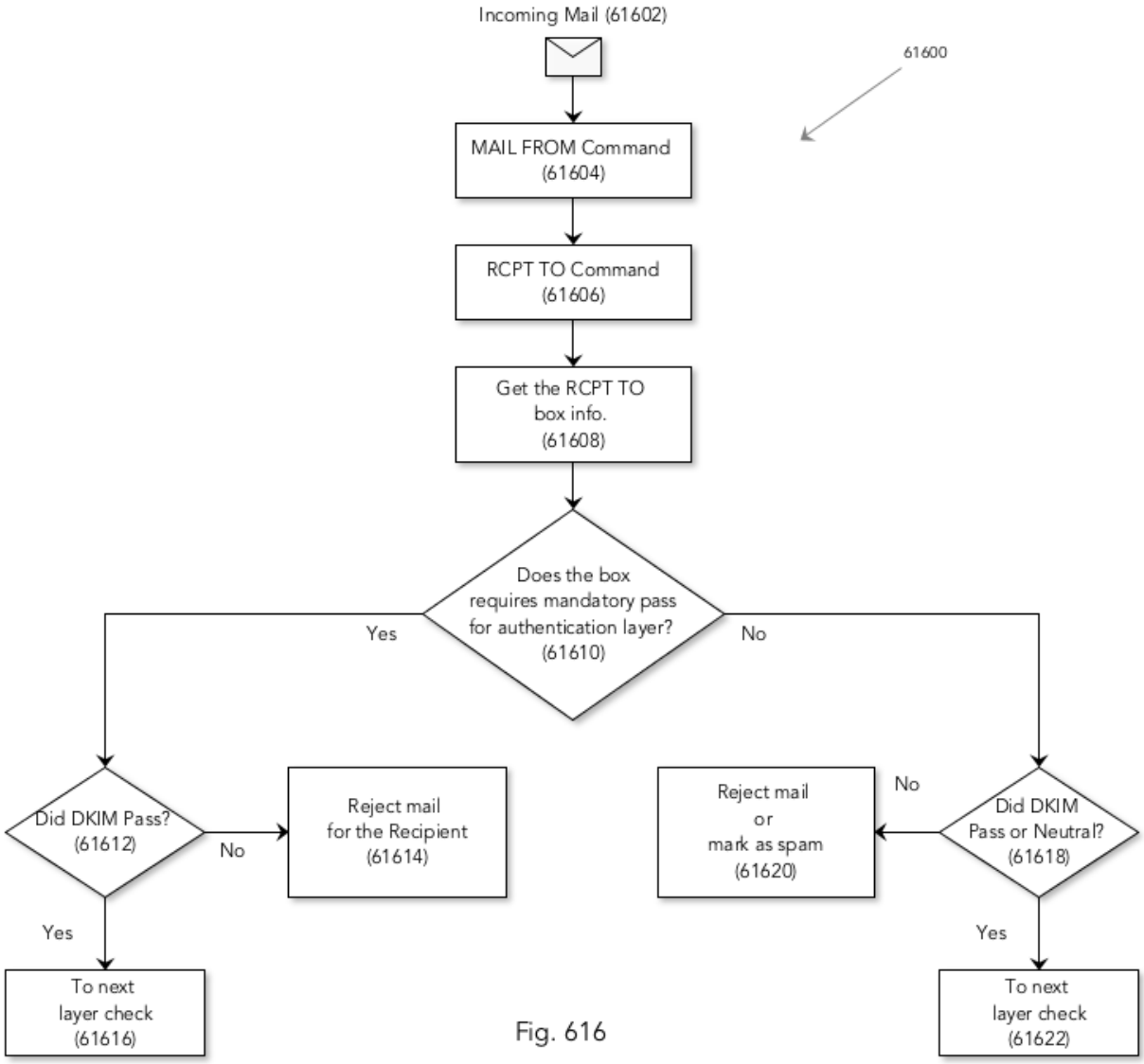

Incoming Mail (61602)
61600
MAIL FROM Command (61604)
RCPT TO Command (61606)
Get the RCPT TO box info. (61608)
Does the box requires mandatory pass for authentication layer? (61610)
Yes
No
Did DKIM Pass? (61612)
No
Reject mail for the Recipient (61614)
Yes
To next layer check (61616)
Reject mail or mark as spam (61620)
No
Did DKIM Pass or Neutral? (61618)
Yes
To next layer check (61622)

Fig. 616

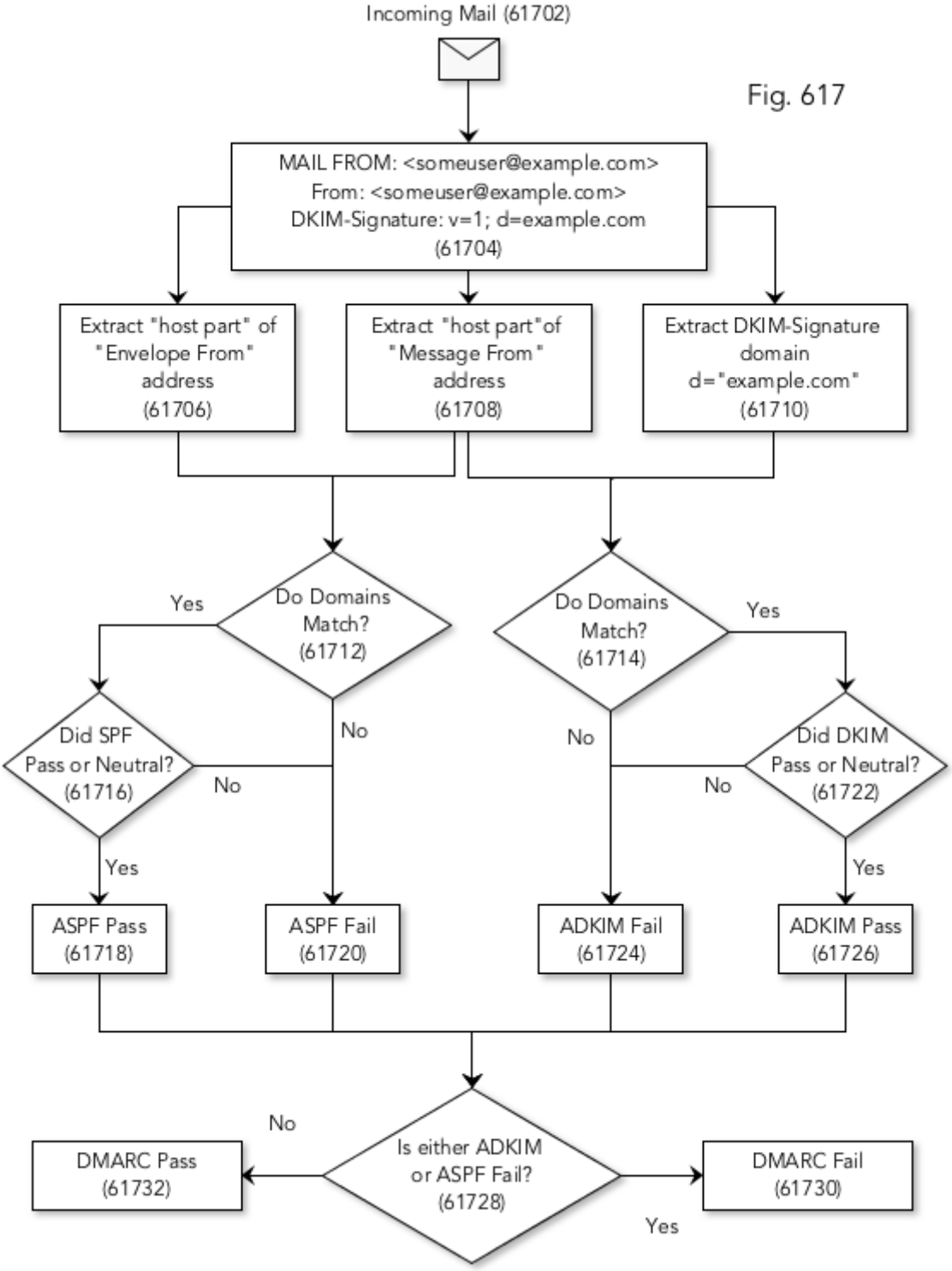

Incoming Mail (61702)
MAIL FROM: <someuser@example.com>
From: <someuser@example.com>
DKIM-Signature: v=1; d=example.com
(61704)
Extract "host part" of "Envelope From" address (61706)
Extract "host part"of "Message From" address (61708)
Extract DKIM-Signature domain d="example.com" (61710)
Do Domains Match? (61712)
Yes
No
Did SPF Pass or Neutral? (61716)
No
Yes
ASPF Pass (61718)
ASPF Fail (61720)
Do Domains Match? (61714)
Yes
No
Did DKIM Pass or Neutral? (61722)
No
Yes
ADKIM Fail (61724)
ADKIM Pass (61726)
Is either ADKIM or ASPF Fail? (61728)
No
DMARC Pass (61732)
DMARC Fail (61730)
Yes

Fig. 617

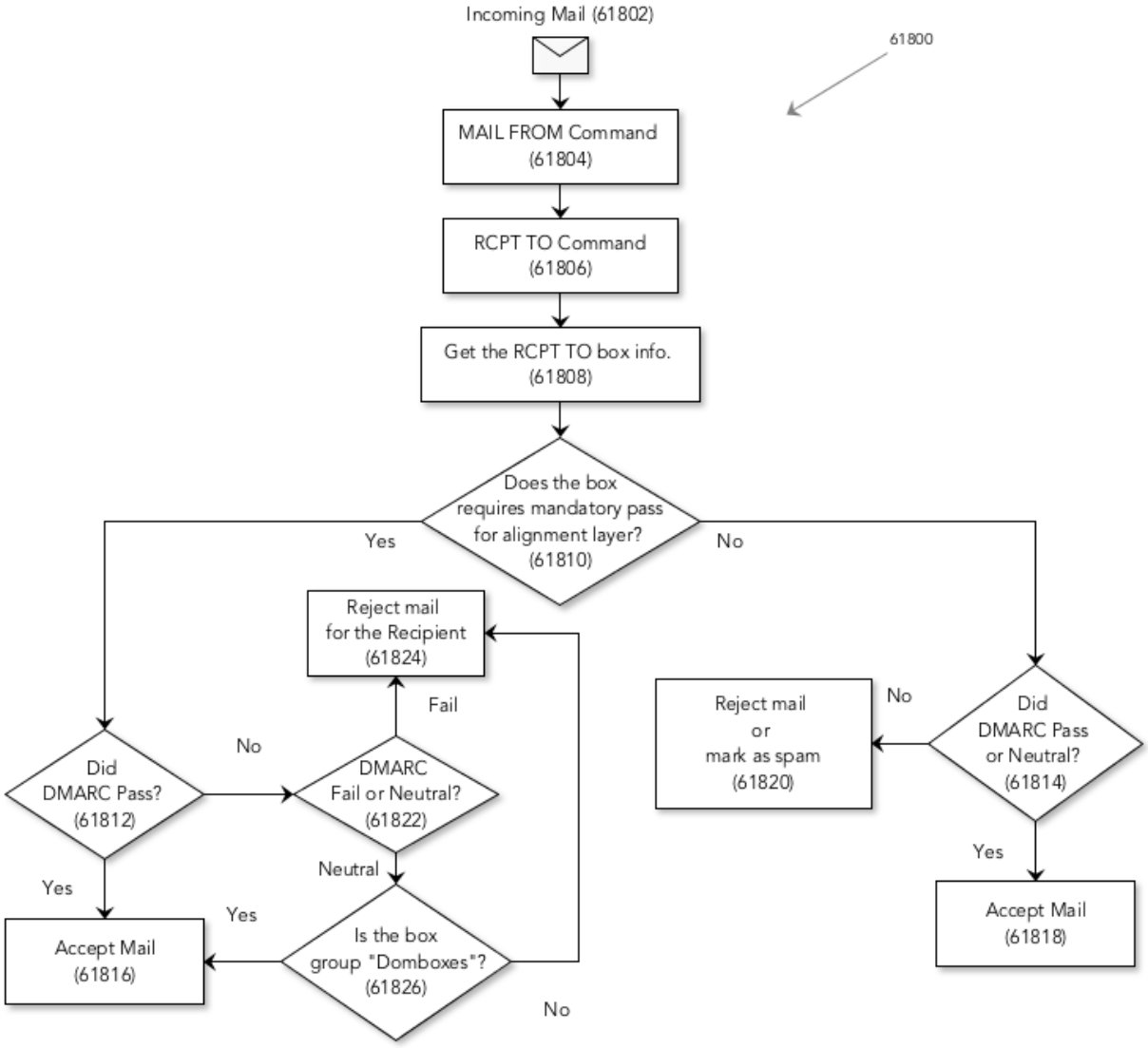

Incoming Mail (61802)
61800
MAIL FROM Command (61804)
RCPT TO Command (61806)
Get the RCPT TO box info. (61808)
Does the box requires mandatory pass for alignment layer? (61810)
Yes
No
Reject mail for the Recipient (61824)
Fail
Did DMARC Pass? (61812)
No
DMARC Fail or Neutral? (61822)
Neutral
Yes
Accept Mail (61816)
Yes
Is the box group "Domboxes"? (61826)
No
Reject mail or mark as spam (61820)
No
Did DMARC Pass or Neutral? (61814)
Yes
Accept Mail (61818)

Fig. 618

FIG. 601 illustrates mail session structure.

A mail session can have unlimited messages **60102**. Each message can have unlimited recipients **60104**.

MAIL FROM1 to MAIL FROMn **60102** command represents beginning of each message. RCPT TO1 to RCPT TOn **60104** command represent recipients of that particular message.

FIG. 602 illustrates a simple SMTP conversation between two mail servers.

mail.example.com is connecting to mail.domboxmail.com with its IP address **60202**. This process known as TCP handshake. The "Client IP" (Mail Sending Server IP) address is extracted from here.

The C letter in FIG. 602 represents the Client (Mail Sending Server). In our case this is mail.example.com

The S letter in FIG. 602 represents the Server (Mail Receiving Server). In our case this is mail.domboxmail.com

The Server responds with 220 code if the server is ready.

The Client issues the HELO command to identify itself.

The Server responds with 250 code to acknowledge.

The Client issues the MAIL FROM **60204** command to specify the sender. This command tells that a new mail transaction is being started. The email address provided by the MAIL FROM command is also known as Envelope From, Return Path, RFC.5321 From and Bounce Address.

The Server responds with 250 code, if there is no problem with the sender address.

The Client issues the RCPT TO **60206** command to specify the receiver mail address. The mail will be delivered to the email address provided in this command. The Client may issue RCPT TO command multiple times to deliver the mail to more than one recipient.

The Server responds with 250 code for each RCPT TO command if the recipient is valid.

The Client issues the DATA command to transfer the message contents (body text, attachments etc)

The Server responds with 354 code to proceed to transfer the message contents.

The Client transfers the message contents (mail headers, body text, attachments etc). The whole contents transferred here is called "Message Part". The headers found in the message contents are called "Message Headers". The "From" header **60210** found in the message header is called "Message From". It is also known as "RFC.5322 From" and "Display From".

The Server responds with 250 code and queue the message for delivery **60214**.

The Client issues the MAIL FROM command again if there is more mails to transfer, otherwise it issues the QUIT command to close the connection.

The Server responds with 221 code and closes the connection.

FIG. 603 illustrates the incoming mail check layers.

Each and every incoming mail has to go through five layer of checks. Those five layers are Encryption Layer **60302**, Authorization Layer **60304**, Alias Layer **60306**, Authentication Layer **60312** and Alignment Layer **60314**. Alias Layer contains two sub layers. Envelope Layer **60308** and Message Layer **60310**

Spam mails usually get caught in one of those layers. So the mail will be rejected instantly.

FIG. 604 illustrates the logical flow of TLS.

An incoming mail **60402** from the Mail sending server (Client) begins the TCP handshake **60404** first. Once this is done, a new SMTP session is created. In this SMTP session we have a global boolean variable called "InTLS". By default InTLS state is set to false **60406**.

Mail sending server (Client) issues the EHLO command. The Mail receiving server (Server) responds with 250 code. The server also provides list of supported SMTP extensions. e.g 250-SIZE, 250-PIPELINING, 250-STARTTLS etc. If STARTTLS extension found in the supported SMTP extensions, then the server supports encryption. So the Mail sending server (Client) issues the STARTTLS **60408** command to encrypt the conversation. The Mail receiving server (Server) responds with 220 code to go ahead.

Both Mail sending server (Client) and Mail receiving server (Server) negotiates the TLS **60410** and then upgrades the current connection to a secure connection. Session "InTLS" state is set to true **60412**. The rest of the SMTP conversations are now encrypted **60414**.

FIG. 605 illustrates the logical flow of Encryption layer check. This layer checks whether the mail is encrypted or not.

An incoming mail **60502** from the sender begins the MAIL FROM **60504** command. At this stage "InTLS" state can be either true or false. If the mail session is upgraded to TLS, then the "InTLS" state is true. The logical flow of upgrading the connection to a secure connection already illustrated in Fig. 604.

Mail Sending Server (Client) issues the RCPT TO **60506** command.

e.g. RCPT TO:<example.com@giri123.domboxmail.com> **60206**

At this stage, we pull the box type for the RCPT TO email address **60508**. In our case, we pull the "example.com@giri123.domboxmail.com" address box type.

If the box type of that address is either Hybrid (H) or Combox (C), then this box requires "mandatory pass" for Encryption layer. In other words "InTLS" state must be "true" to proceed. So at this stage, we check whether the box requires encryption or not **60510**.

If the box doesn't require encryption, then we can proceed to other layer checks **60512**.

If the box require encryption, then we check the "InTLS" state **60514**. If the state is "true", then the Encryption layer check is passed. So we proceed to other layer checks **60516**. If the state is "false", that means the current mail is being transferred as unencrypted plain text. At this point we reject the mail for that recipient **60518**.

If there is more than one recipient, we repeat the same encryption layer check for every RCPT TO command. i.e. For each and every mail Recipient.

FIG. 606 illustrates the logical flow of SPF. This layer checks whether the mail sending server (Client) is authorized to carry the message or not. This check is processed for each and every MAIL FROM **60102** command.

An incoming mail **60602** from the sender begins the TCP handshake **60604**. This is the same TCP handshake **60404** happened in Encryption layer. Mail sending server IP address (Client IP) is extracted at this stage and stored in a global session variable called CLIENT_IP **60606**. e.g CLIENT_IP = xxx.xxxx.xxx.xxx.

Mail Sending Server (Client) issues the MAIL FROM **60608** command. e.g. MAIL FROM:<john@example.com>

Get the "domain part" from the MAIL FROM address **60610**. In our case this is example.com

Fetch SPF Record of the "domain part" from the DNS server **60612**. In our case example.com DNS server. The record returned from the DNS would look something like this. “v=spf1 ip4:xxx.xxx.xxx.xxx ip4:yyy.yyy.yyy.yyy +a +mx -all”

If there is no SPF Record on the DNS, that means the domain owner have not configured any SPF. In this case SPF result is NEUTRAL.

If there is SPF Record on the DNS, then we check whether the CLIENT_IP found in the list of IP addresses found in the SPF record **60614**.

If the CLIENT_IP found in the list of whitelisted IP addresses, that means "SPF Pass". **60616**.

If the CLIENT_IP not found in the list of whitelisted IP addresses, that means "SPF Fail". **60618**.

FIG. 607 illustrates the logical flow of Authorization layer check.

An incoming mail **60702** from the sender begins the MAIL FROM **60704** command. At this stage SPF is checked and the result is stored in the global message **60102** variable.

Mail Sending Server (Client) issues the RCPT TO **60706** command. e.g. RCPT TO:<example.com@giri123.domboxmail.com>

At this stage, we pull the box type for the RCPT TO email address **60708**. In our case, we pull the "example.com@giri123.domboxmail.com" address box type.

If the box type of that address is either Hybrid (H) or Combox (C), then this box requires "mandatory pass" for Authorization layer **60304**. So at this stage, we check whether the box requires mandatory pass or not for authorization layer **60710**.

If the box require "mandatory pass" for authorization layer, then we check whether SPF has passed or not **60712**. If SPF has passed then we can proceed to the next layer check **60716**. If SPF has not passed then we reject the mail for the recipient **60714**.

If the box doesn't require "mandatory pass" for authorization layer, then we check whether SPF has passed or neutral **60718**. If SPF has passed or Neutral then we can proceed to the next layer check **60722**. If SPF has failed then we can either reject mail or mark the mail as spam **60720**.

FIG. 608 illustrates the logical flow of SAD.

BuyFruits.com **60802** is a company that sells fruits. This is the parent company. But it also has three subsidiaries BuyOranges.com **60810**, BuyApples.com **60812**, BuyGrapes.com **60814**

When a user create Dombox for BuyOranges.com, the dombox address will be buyoranges.com@giri123.domboxmail.com **60804**. For this Dombox, only the mails from BuyOranges.com are allowed.

When a user create Dombox for BuyApples.com, the dombox address will be buyapples.com@giri123.domboxmail.com **60806**. For this Dombox, only the mails from BuyApples.com are allowed.

When a user create Dombox for BuyGrapes.com, the dombox address will be buygrapes.com@giri123.domboxmail.com **60808**. For this Dombox, only the mails from BuyGrapes.com are allowed.

There should be a way for the parent company to send mails to subsidiary company users. e.g. A mail from the parent company CEO (ceo@buyfruits.com) to the subsidiary company users.

We solve this problem with our standard called Sender Alias Domains (SAD). SAD is applicable only for Domboxes **40104**.

SAD should be placed in the "Dombox Domain". buyapples.com@giri123.domboxmail.com where buyapples.com is the dombox domain. So the SAD should be placed in buyapples.com

The SAD structure would look like this. v=sad1 buyfruits.com -all

If the above string found in buyapples.com DNS record **60818**, that would allow mails from @buyfruits.com to the Dombox buyapples.com@giri123.domboxmail.com. In other words, although the box created only for buyapples.com, it can receive mails from buyfruits.com.

If SAD records not found in DNS, then we allow only the Dombox Domain "buyapples.com" to send email to the user.

In Fig. 608 "v=sad1 buyfruits.com -all" **60818** SAD Record is defined in all three subsidiary domains DNS **60816**. i.e. BuyOranges.com **60810**, BuyApples.com **60812**, BuyGrapes.com **60814**. So buyfruits.com **60802** now can send mails to subsidiary domain users. Because buyfruits.com is a "Sender Alias Domain" for the subsidiary domains.

In Fig. 608 dotted arrows represents indirect mail delivery. i.e. via a Sender Alias Domain.

As of now the SAD Records are duplicated in subsidiary domains. i.e. The same SAD Record present in all three subsidiary domain DNS. SAD Record can be managed in only one place with the help of "redirect" tag.

We can place the main SAD Record "v=sad1 buyfruits.com -all" in _sad.buyfruits.com and then in all subsidiary domains we can use the following SAD Record. "v=sad1 redirect:_sad.buyfruits.com". Now all the subsidiary SAD queries are redirected to _sad.buyfruits.com. If we add more domains in the future, we don't have to edit each and every subsidiary domain. We have to only edit the main SAD.

We can also have "include" tag. This will include the external SAD. "v=sad1 example1.com -all" Record is placed in _sad.abc.com and "v=sad1 example2.com -all" Record is placed in _sad.xyz.com.

Now we can use the "include" tag to include those SAD. "v=sad1 include:_sad.abc.com include:_sad.xyz.com -all". If that SAD found in a domain, that would allow both example1.com and example2.com. There is a maximum of 10 DNS lookups in order to avoid DDoS attacks.

FIG. 609 illustrates the logical flow of Alias - Envelope layer - Fake Pass check.

Mail Sending Server (Client) issues the RCPT TO **60902** command. e.g. RCPT TO:<giri@domboxmail.com>

At this stage, we pull the box type for the RCPT TO email address **60904**. In our case, we pull the "giri@domboxmail.com" address box type.

We check **60906** whether the box is a box found in "Domboxes" **40104** group. If the box type of that address is either Primary (P) or Mailbox (M), then this is a box found in "Mailboxes" **40102** group. If the box type of that address is either Dombox (D), Hybrid (H) or Combox (C), then this is a box found in "Domboxes" **40104** group

If the box is a box found in Domboxes **40104** group, then we proceed to the "Envelope SAD Direct Pass" Check **60908**.

If the box is a box found in Mailboxes **40102** group, then we set the Alias - Envelope layer main result state to "PASS" and sub state to "FAKEPASS" **60910** and then proceed to next layer check **60912**.

This is because we will be having mail score from 1 to 5. i.e. For each layer, if the result is "PASS", the score is 1 will be applied. Mail Score will be explained in the next chapter. SAD is not applicable for Mailboxes **40102**. So we use "PASS" with "FAKEPASS" for Alias Layer in the box found in Mailboxes **40102** to keep the score consistent.

FIG. 610 illustrates the logical flow of Alias - Envelope layer - Direct Pass check.

Get "host part" of "MAIL FROM" command **61002**. e.g. example.com <= MAIL FROM:<john@example.com>

Get "dombox domain" of "RCPT TO" address **61004**. e.g. example.com <= RCPT TO:<example.com@giri123.domboxmail.com> [Note: At this point we also pull the SAD record from the “dombox domain” and store it in our session cache]

Is the "Dombox Domain" matches with "MAIL FROM" domain? **61006**.

If yes, we mark the "Alias - Envelope Layer" as Direct Pass **61010** and then proceed to next layer check **61012**. Because the mail sending domain is the dombox domain. We set the Alias - Envelope layer main result state to "PASS" and sub state to “DIRECTPASS"

If no, then we proceed to the "Envelope SAD Indirect Pass" Check **61008**.

FIG. 611 illustrates the logical flow of Alias - Envelope layer - Indirect Pass check.

Get "host part" of "MAIL FROM" command **61102**. e.g. buyfruits.com <= MAIL FROM:<john@buyfruits.com>

Get "dombox domain" of "RCPT TO" address **61104**. e.g. buyapples.com <= RCPT TO:<buyapples.com@giri123.domboxmail.com>

Is the "Dombox Domain" matches with "MAIL FROM" domain? **61106**.

If yes, we mark the "Alias - Envelope Layer" as Direct Pass **61108**

If no, we fetch the SAD Record from "Dombox Domain" DNS. In our case buyapples.com DNS **61110**. DNS response would look like this. "v=sad1 buyfruits.com -all"

Is SAD Record found in DNS TXT Records? **61112**

If no, Reject mail for the recipient **61114**.

If yes, check whether the "MAIL FROM" domain present in the SAD Record **61116**. In our case we check whether buyfruits.com present in the SAD Record.

SAD Record would look like this. "v=sad1 buyfruits.com:r+b -all"

Get the "Envelope From" domain host configuration from the SAD Domains In our case, Host => buyfruits.com Config => Relaxed Mode (r) and Both Mode (b)

Check whether the "MAIL FROM" domain is allowed in the “Alias - Envelope Layer". In our case we check whether buyfruits.com has envelope mode (e) or both mode (b).

If no, that means the sender is not allowed to send mails to the dombox. Mark the Layer as "Envelope SAD Fail" **61118** and then reject mail for the recipient **61120**.

If yes, that means the sender is allowed to send mails to the box. Mark the Layer as "Envelope SAD Indirect Pass" **61122** and then proceed to next layer check **61124**. We set the Alias - Envelope layer main result state to "PASS" and sub state to "INDIRECTPASS"

FIG. 612 illustrates the logical flow of Alias - Message layer - Fake Pass check.

Mail Sending Server (Client) issues the RCPT TO **61202** command. e.g. RCPT TO:<giri@domboxmail.com>

At this stage, we pull the box type for the RCPT TO email address **61204**. In our case, we pull the "giri@domboxmail.com" address box type.

We check whether the box is a box found in "Domboxes" **40104** group. If the box type of that address is either Primary (P) or Mailbox (M), then this is a box found in "Mailboxes" **40102** group. If the box type of that address is either Dombox (D), Hybrid (H) or Combox (C), then this is a box found in "Domboxes" **40104** group.

If the box is a box found in Domboxes **40104** group, then we proceed to the "Message SAD Direct Pass" Check **61208**.

If the box is a box found in Mailboxes **40102** group, then we set the Alias - Message layer main result state to "PASS" and sub state to "FAKEPASS" **61210** and then proceed to next layer check **61212**. This is because SAD is not applicable to Mailboxes.

FIG. 613 illustrates the logical flow of Alias - Message layer - Direct Pass check.

Get "host part" of "Message From" header **61302**. e.g. example.com <= From:<john@example.com>

Get "dombox domain" of "RCPT TO" address **61304**. e.g. example.com <= RCPT TO:<example.com@giri123.domboxmail.com>

Is the "Dombox Domain" matches with "Message From" header domain? **61306**.

If yes, we mark the "Alias - Message Layer" as Direct Pass **61310** and then proceed to next layer check **61312**. Because the “Message From” domain is the “Dombox Domain”. We set the “Alias - Message layer” main result state to "PASS" and sub state to "DIRECTPASS"

If no, then we proceed to the "Message SAD Indirect Pass" Check **61308**.

FIG. 614 illustrates the logical flow of Alias - Message layer - Indirect Pass check.

Get "host part" of "Message From" header **61402**. e.g. buyfruits.com <= From: John Doe <john@buyfruits.com>

Get "dombox domain" of "RCPT TO" address **61404**. e.g. buyapples.com <= RCPT TO:<buyapples.com@giri123.domboxmail.com>

Is the "Dombox Domain" matches with "Message From" header domain? **61406**.

If yes, we mark the "Alias - Message Layer" as Direct Pass **61408**

If no, we fetch the SAD Record of "Dombox Domain" from the session cache (We already fetched the SAD for Envelope SAD check.). In our case buyapples.com **61410**. SAD Record would look like this. "v=sad1 buyfruits.com:r+b -all"

Get the "Message From" domain host configuration from the SAD Domains **61412**. In our case, Host => buyfruits.com Config => Relaxed Mode (r) and Both Mode (b)

Check whether the "Message From" domain is allowed in “Alias - Message Layer" **61414**. In our case we check whether buyfruits.com has message mode (m) or both mode (b).

If no, that means the sender is not allowed to use the address for "Message From". Mark the Layer as "Message SAD Fail" **61416** and then reject mail for the recipient **61418**.

If yes, that means the sender is allowed to use the address for "Message From". Mark the Layer as "Message SAD Indirect Pass" **61420** and then proceed to next layer check **61422**. We set the “Alias - Message layer” main result state to "PASS" and sub state to "INDIRECTPASS"

FIG. 615 illustrates the logical flow of DKIM.

Extract DKIM-Signature "Message Header" from the "Message Part" **61504**.

If no DKIM-Signature found that means DKIM result is NEUTRAL.

DKIM-Signature: v=1; a=rsa-sha256; s=dev; d=example.com; c=simple/simple; q=dns/txt; h=Received:From:To:Subject:Date:Message-ID; bh={body hash goes here}; b={signature value};

Get the "d" (domain) tag value. => example.com

Get the "s" (selector) tag value => dev

Fetch the DKIM public key from the TXT record found in this path. {s tag value}.domainkey.{d tag value} **61506**.

i.e. dev._domainkey.example.com

Verify the DKIM-Signature using the public key **61508**.

Is verification passed? **61510** If yes DKIM Pass **61512**. If no, DKIM Fail **61514**

FIG. 616 illustrates the logical flow of Authentication layer check.

An incoming mail **61602** from the sender begins the MAIL FROM **61604** command.

Mail Sending Server (Client) issues the RCPT TO **61606** command. e.g. RCPT TO:<example.com@giri123.domboxmail.com>

At this stage, we pull the box type for the RCPT TO email address **61608**. In our case, we pull the "example.com@giri123.domboxmail.com" address box type.

If the box type of that address is either Hybrid (H) or Combox (C), then this box requires "mandatory pass" for Authentication layer. So at this stage, we check whether the box requires mandatory pass or not for authentication layer **61610**.

If the box require "mandatory pass" for authentication layer, then we check whether DKIM passed or not **61612**. If DKIM has passed then we can proceed to the next layer check **61616**. If DKIM has not passed then we reject the mail **61614**.

If the box doesn't require "mandatory pass" for authentication layer, then we check whether DKIM passed or not **61618**. If DKIM has passed or Neutral then we can proceed to the next layer check **61622**. If DKIM has not passed then we can either reject mail or mark the mail as spam **61620**.

FIG. 617 illustrates the logical flow of DMARC. An incoming mail **61702** with "Message From" address "someuser@example.com" **61704** is checked with DMARC.

Get "host part" of "Message From" **60210** header. e.g. example.com <= From:John <someuser@example.com>

Check whether a valid DMARC record exists in "Message From" **60210** domain in this path. _dmarc.{Message From Domain}. e.g. _dmarc.example.com

If a valid record not exists, then DMARC result is NEUTRAL

Get the "Host Part" from "Envelope From". i.e RFC5321.From **61706**

Get the "Host Part" from "Message From". i.e RFC5322.From **61708**

Get the "d" (domain) tag value of DKIM-Signature. **61710**

Is the RFC5322.From "Host Part" matches RFC5321.From "Host Part"? **61712**

If No, SPF is not Aligned **61720**

If Yes, Check for SPF Result. **61716**

If Pass or Neutral, SPF is Aligned (ASPF Pass). **61718**

If Fail, SPF is not Aligned (ASPF Fail) **61720**

If the RFC5322.From "Host Part" matches DKIM-Signature "d" (domain) tag value? **61714**

If No, DKIM is not Aligned. **61724**

If Yes, Check for DKIM Result. **61722**

If Pass or Neutral, DKIM is Aligned (ADKIM Pass). **61726**

If Fail, DKIM is not Aligned (ADKIM Fail). **61724**

Is either ADKIM or ASPF Fail? **61728**

If Yes, DMARC Fail. **61730**

If No, DMARC Pass. **61732**

FIG. 618 illustrates the logical flow of Alignment layer check.

An incoming mail **61802** from the sender begins the MAIL FROM **61804** command.

Mail Sending Server (Client) issues the RCPT TO **61806** command. e.g. RCPT TO:<example.com@giri123.domboxmail.com>

At this stage, we pull the box type for the RCPT TO email address **61808**. In our case, we pull the "example.com@giri123.domboxmail.com" address box type.

If the box type of that RCPT TO address is either Hybrid (H) or Combox (C), then this box requires "mandatory pass" for Alignment layer. So at this stage, we check whether the box requires mandatory pass or not for alignment layer **61810**.

If the box require "mandatory pass" for alignment layer, then we check whether DMARC passed or not **61812**. If DMARC has passed then we can proceed to accept the mail **61816**. If DMARC has not passed then we check whether the DMARC result is Fail or Neutral **61822**. If the DMARC result is "Fail", then we reject the mail for the recipient **61824**. If the DMARC result is "Neutral", then we check whether the box group is "Domboxes" **61826**. If the box group is "Domboxes" then we accept the mail **61816**. If not we reject the mail **61824**.

If the box doesn't require "mandatory pass" for alignment layer, then we check whether DMARC passed or neutral **61814**. If DMARC has passed or Neutral then we can proceed to accept the mail **61818**. If DMARC has not passed then we can either reject mail or mark the mail as spam based on the policy provided by the DMARC Record **61820**.

The Alignment Layer **60314** result for mails found in Domboxes **40104** group most likely will be "Pass" unless DMARC result is "FAIL".

The Alignment Layer **60314** deals with the "Message From" **60210** domain. This layer relies on DMARC standard. In Domboxes **40104**, if the DMARC result is Fail, then most likely the mails will be rejected. If the DMARC is Neutral, that means the "Message From" domain doesn't have any DMARC record in their DNS. Since the Domain is already Alias Layer passed, we can mark this layer as "Pass". This is because the website owner says that he/she takes responsibility for the mail via Alias Layer. So the Alignment Layer **60314** result will be most likely a “Pass" for Domboxes.

The Encryption Layer **60302**, checks whether the incoming message is encrypted or not. This layer uses TLS protocol.

Encryption Layer Result Main state can be either PASS or FAIL.

There is no sub state available for Encryption Layer.

The Authorization Layer **60304**, checks whether the mail sending server is authorized to carry the message or not for the envelope domain . This layer uses a standard called Sender Policy Framework (SPF).

Authorization Layer Result Main state can be either PASS, NEUTRAL or FAIL.

NEUTRAL state can have one of the following sub states: NONE, NEUTRAL

FAIL state can have one of the following sub states: FAIL, SOFTFAIL, TEMPERROR, PERMERROR

The Alias Layer **60306**, checks whether the "Envelope From" domain and "Message From" domain are authorized alias for the "Dombox Domain". This layer uses our proprietary standard called Sender Alias Domains (SAD). SAD will be explained in later section. This layer contains two sub layers. Envelope Layer **60308** and Message Layer **60310**.

The Alias - Envelope Layer **60308**, checks whether the "Envelope From" domain is an authorized alias for "Dombox Domain".

The Alias - Envelope Layer Result main state can be one in the following. PASS or FAIL

PASS state can have one of the following sub states: FAKEPASS, DIRECTPASS, INDIRECTPASS

The Alias - Message Layer **60310**, checks whether the "Message From" domain is an authorized alias for "Dombox Domain".

The Alias - Message Layer Result main state can be one in the following. PASS or FAIL

PASS state can have one of the following sub states: FAKEPASS, DIRECTPASS, INDIRECTPASS

The overall Alias Layer **60306** result depends on the sub layer results. If one sub layer result is fail, then the overall Alias Layer **60306** result is Fail.

Note: Alias Layer can have only "PASS" result. When the layer result is "FAIL", mail will be rejected. Mail will be accepted only in development/testing mode.

The Authentication Layer **60312**, checks whether the mail is digitally signed by the sending server. This layer uses the standard DomainKeys Identified Mail (DKIM).

Authentication Layer Result main state can be one in the following. PASS, NEUTRAL or FAIL.

NEUTRAL state can have the following sub state: NONE

FAIL state can have one of the following sub states: FAIL, TEMPERROR, PERMERROR

The Alignment Layer **60314**, checks whether the "Message From" domain is aligned properly with SPF Domain and DKIM domain. If not aligned, then it applies the policy fetched from the "Message From" domain. This layer uses a standard called "Domain-based Message Authentication, Reporting and Conformance (DMARC)".

Alignment Layer Result main state can be one in the following. PASS, NEUTRAL or FAIL.

There is no sub state available for Alignment Layer.