Consiglio Nazionale delle Ricerche

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"

ISTI Technical Reports

Definition of a new model of communication: Secure Application Email (SAE)

Francesco Gennai, ISTI-CNR, Pisa, Italy

Fabio Sinibaldi, ISTI-CNR, Pisa, Italy

Marina Buzzi, IIT-CNR, Pisa, Italy

Loredana Martusciello, IIT-CNR, Pisa, Italy

ISTI-TR-2023/002

Consiglio Nazionale delle Ricerche

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"

Definition of a new model of communication: Secure Application Email (SAE)

F. Gennai, F. Sinibaldi, M. Buzzi, L. Martusciello

ISTI-TR-2023/002

In this technical report, we define a Secure Application Email model and protocol that works on top of existing Internet email architecture that can be used in the development of new services with enanched security. The new Secure Application Email model could represent an evolution of the current Internet email model while keeping a deep level of interoperability between the two models.

Keywords: email, Secure Email, SMTP, Message Transfer Agent, DNSSEC, Cryptography, Mail relay.

Citation

Gennai F., Sinibaldi F., Buzzi M., Martusciello L. *Definition of a new model of communication: Secure Application Email (SAE).* ISTI Technical Reports 2023/002. DOI: 10.32079/ISTI-TR-2023/002.

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"
Area della Ricerca CNR di Pisa
Via G. Moruzzi 1
56124 Pisa Italy
http://www.isti.cnr.it

Definition of a new model of communication: Secure Application Email (SAE)

Francesco Gennai[1], Fabio Sinibaldi[1], Loredana Martusciello[2], Marina Buzzi[2]
[1] Institute of Information Science and Technology - ISTI, National Research Council of Italy - CNR
[2] Institute for Informatics and Telematics - IIT, National Research Council of Italy - CNR
francesco.gennai@isti.cnr.it, fabio.sinibaldi@isti.cnr.it,
loredana.martusciello@iit.cnr.it, marina.buzzi@iit.cnr.it

February 2023

Abstract.
In this technical report, we define a Secure Application Email model and protocol that works on top of existing Internet email architecture that can be used in the development of new services with enanched security. The new Secure Application Email model could represent an evolution of the current Internet email model while keeping a deep level of interoperability between the two models.

Introduction.
A network application service is defined through its functions and through its communication protocol.
In a layer architecture, the communication protocol, thus also the service, can be seen as a user of the service that the underlying communication architecture provides.
Thanks to a careful and proper definition of communication protocol of the service, it could exploit the capabilities of the selected communication architecture for transporting information end-to-end.
For the Secure Application Email (SAE) model, we can identify the underlying communication architecture selected for its functioning in the Internet email system. Figure 1 shows the architecture of the SAE model.

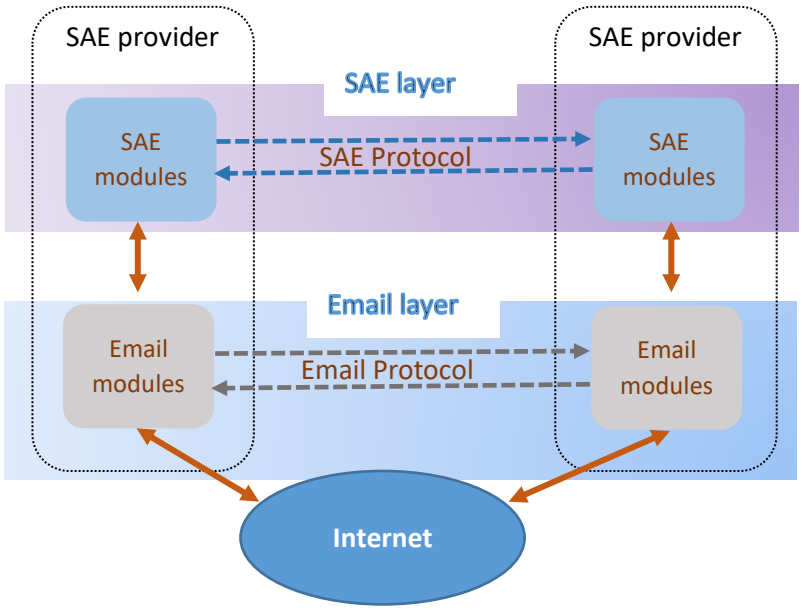


Figure 1 - Secure Application Email Model

The SAE model is particularly suitable for defining systems that, while using Internet email, need special features that Internet email does not offer. These include, for example, Certified Electronic Email (CEM) systems.
As we will see in this report, SAE promotes better definition and control of events related to the transport of an email within the Internet system.

SAE provider certification.

SAE introduces the concept of Email Authority, which present analogies with that of Certification Authority. An Email Authority can declare the domains it manages through mechanisms that make the declaration reliable.

As a result, it is possible to identify with a high degree of reliability the Message Transfer Agents of the Email Internet system, which belong to an Email Authority of an SAE infrastructure and operate in compliance with the relative functioning rules.

The identification of an SAE domain is done by certifying that it belongs to an SAE provider.

One of the most critical aspects in such an architecture is the identification of the MTA authorized to manage a domain.

This identification is done through a query to DNS to obtain an MX record. Without DNSSEc/DANE, the DNAS response cannot be considered reliable.

While providers are able to adopt the technologies and operative modalities to comply with the required security requirements, for the owner of an email domain compliance with the necessary security requirements may not be as easy; in fact, the email domain may be managed by an organization outside the provider, which delegates to the provider the management of the service through the appropriate configuration of its name server (insertion of the MX record in the DNS zone of the domain). Thus, reliability will depend on the infrastructure of the domain owner and not that of the provider to whom management of the email domain is delegated. Consider, for example, a classic scenario in which the delegation of the management of an email domain to a provider is done through an MX record placed in the DNS zone managed by an organization other than the provider. In this case, it might be difficult to force this organization to adopt technologies such as DNSSEC, which are necessary to ensure the appropriate level of security. Not only that, the provider may have weak control over the level of operational quality with which the outside organization manages its name server, thus reducing the level of assurance about the email service offered.

It is therefore a question of seeing how the Internet email system can meet these particular reliability requirements.

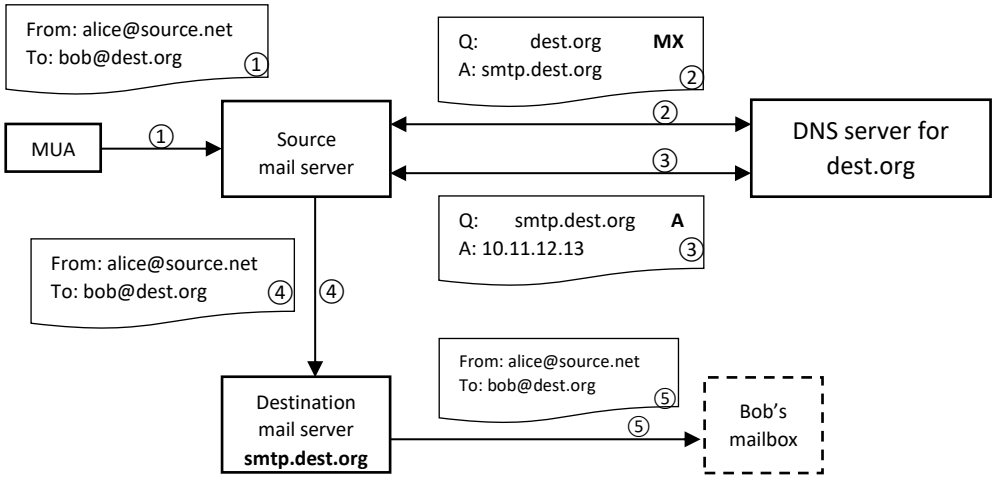


Figure 2 - SMTP Protocol—A client sends outgoing mail by connecting to its organization's local SMTP server (❶). The local server performs a DNS lookup for the mail exchange (MX) record of the destination.com domain, which contains the hostname of the destination's SMTP server, in this case **smtp.dest.org** (❷). The sender's server then performs a second DNS lookup for the destination server's IP address (❸), establishes a connection, and relays the message (❹). The message is delivered to the recipient mailbox (❺).

Layer architecture, separation of layers.
For the study or design of any communication system, it is propaedeutic to be able to identify and separate its functional levels. Each layer has its own functional characteristics, which must be known and interpreted for a better definition of the higher layer that exploits it for its communications. A layer may, for example, have features that prioritize efficiency over reliability. This is not a problem for a complete communication architecture if the tasks not performed by the layer under consideration are performed by the higher layer, in case they are deemed necessary for a particular communication system. Referring to some of the fundamental concepts of a communication architecture, we can identify some of the characteristics that a given layer may have:

- connectionless
- connection oriented
- flow control
- service point address

For example, in TCP/IP, the IP layer is connectionless and has routing capabilities for data delivered to it from the upper layer.
TCP, which exploits the capabilities of the IP layer, is connection oriented. An application that needs to "open" a connection on which to transmit/receive an ordered data stream with its remote correspondent can make use of TCP. An application that does not need coordinated and ordered connection management may use the UDP protocol.
It is important to note that an application that uses the UDP protocol could have the data flow control functions within it, making it, in functional terms, analogous to an application that does not have data flow control within it, but exploits the TCP protocol.
Each layer is geared to fulfill a particular function of a communication system.
It is also important to note that each layer presents itself to the higher layer through an address (service point address). For example, ports of the TCP layer represent the addresses by which the TCP layer communicates with the upper layer. The TCP port number is a selector of the application to which TCP should match a given data stream: one can consider this as a feature for selecting or routing data to the respective applications at the ends of a TCP connection.

This brief introduction should help us better understand the SAE architecture proposed in this paper.

The SAE system is, by its definition, a system that uses the Internet e-mail system.
In the introduction we mentioned the characteristics that an SAE system must have, at this point, these characteristics become the goals we also want to achieve through the use of the Internet email system.

The Internet email system.
The Internet email system has specific functioning characteristics that we can exploit for the SAE system. It also has some limitations, among which the critical issues in the certain identification of the server managing a domain (mainly due to the limitations of the Domain Name System without the adoption of DNSSEC) are relevant. Let us do not consider, for the moment, the user identification, which concerns the interface by which the user accesses the service.
The Internet email system also has its own error management and reporting system.

The SAE system.
Among the functioning features of the Internet email system, we can identify some that are of particular interest for the design of an SAE system.

Thus, considering the SAE system as a higher layer that exploits Internet email as a transport sublayer to carry out its communications, we need to identify which features, necessary for SAE, are already present in Internet email in order to exploit them and which are absent from the Internet email system, which therefore will have to be defined and resolved within the SAE layer.
Some authentication and authorization functions are important for the SAE system:
1) Authentication of an SMTP server.
2) Authorization of an SMTP server to operate as an SAE server.
3) Authentication of an email domain.
4) Authorization of a domain to operate as an SAE domain.

Authentication of an SMTP server identifies with certainty the server to which the SMTP client opens a connection to send an email.
In the Internet email system, the SMTP client selects the SMTP server to which to open a connection, through the Domain Name System. The name of the SMTP server is translated into its IP address by a query to the DNS. DNS cannot guarantee that the response is correct, for example, a MITM attack on DNS could change the IP address that is returned to the client. The use of DNSSEC, an extension to DNS, would allow query responses to be trusted. DNSSEC is not yet widespread because of its complexity, so alternative solutions independent of DNSSEC have been found for several network protocols, such as HTTP and SMTP, to verify the validity of the DNS response.
One solution for authenticating an SMTP server, independent of DNSSEC, is the MTA-STS standard (RFC 8461).

Once the SMTP server has been authenticated, it must be verified whether it is authorized to operate as an SAE server.
The SMTP client must open the SMTP session to an authorized SMTP server. The SMTP client can verify the authorization of an SMTP server by comparing the X509 certificate obtained from the SMTP server at the beginning of the SMTP session with a certificate list of SAE authorized servers. Another solution is to provide the SMTP client with the IP address of the authorized SMTP server by using "domain literal" (see RFC 5321), thus avoiding querying DNS. The latter solution is less robust than the former, but is still applicable in cases where the desired level of security allows it.

The authentication of an email domain has similar aspects to the authentication of an email server. In this case, the membership of the email domain in a specific email server must be authenticated. Once again the unreliability of DNS plays its role. In fact, the SMTP client, in order to determine the email server to which an email domain belongs must query the DNS (query for the MX record of the domain) to obtain the name of the email server with which to open the SMTP connection for sending emails destined to that email domain. The adoption of DNSSEC would secure the response to the query for the MX record, but, as mentioned above, DNSSEC is not yet widely adopted. As noted above, the domain could be managed by organizations that do not have the operational capabilities of a large provider and therefore may find it more difficult to adopt new technologies and/or infrastructure for enhanced security.
Again, domain authentication could be achieved with the MTA-STS standard (RFC 8461), a DNSSEC-independent solution.

The authorization of a domain to operate as an SAE can be delegated to a control authority, or delegated to the control that each individual SAE provider can perform.
This function is specific to the SAE system, so it is not present in the Internet email system.

Let's take two MTAs, one sender (MTA-S), the other recipient (MTA-R).

The classic email architecture.
For an email addressed to one or more MTA-R addresses, MTA-S can open a TLS session to transfer a single copy of the email (even if addressed to multiple recipients) to MTA-R (for simplicity, for now, we do not consider temporary or permanent error cases).
TLS is normally used only for session encrypt, not to identify the MTA-R server.
To identify an MTA system as an SAE system, TLS can be used for MTA server identification. In this case it is necessary to provide all MTA-S with access to a list of X509 certificates that identify MTAs belonging to the SAE system. In the case of some solutions, such as Registered Email (REM - standard ETSI 319 532-4) this access is through the Common Service Interface, which includes the Trusted List (an XML file listing providers belonging to a REM circuit and shared among them).

The SAE email architecture.
As in the previous section, we analyze the transition of an email from MTA-S to MTA-R.
The SMTP protocol defines the mechanisms by which an email passes from the client to the server for a destination address. Where there are multiple destination addresses to the same server, for the same email, it suggests (but does not force) that the multiple addresses be merged into a single SMTP session, thus optimizing transmission by transferring a single copy of the email.
Without going into detail, one can imagine that we face a critical point, in that in a "macroscopic" picture of the transfer phase of an email from MTA-S to MTA-R it would be useful to define a single global event representing the actual transfer of the email (with all its destination addresses) to the MTA-R server. However, we have just seen that with the SMTP protocol this behavior is not guaranteed. With the SMTP protocol an email with multiple destination addresses to the same MTA-R might be treated differently (example: of 4 addresses one might receive a temporary error, possibly resulting in the same email being transmitted through two different SMTP sessions , the first for the 3 addresses without error, the second for the address with temporary error).
The SAE system responds simply and efficiently to all those systems, such as CEM systems, that may need to "certify" in a simple and reliable way the various events related to the transport of an email.

Consider again the case where a destination address is invalid at the destination server (MTA-R).
In normal Internet email system operation, this event may be detected during the SMTP session or, later, after the email has been received by MTA-R. In the former case, the handling of the event, would be the responsibility of MTA-S, although it is an event caused by MTA-R.
Taking again as an example a CEM system, where, as mentioned above may be required to certify the passing of an email from MTA-S to MTA-R, and considering that the different negative events that occur during the SMTP session are caused by MTA-R but detected and certified by MTA-S, MTA-S would find itself certifying an event caused by MTA-R thus creating a criticality for a possible verification of responsibilities between MTA-S and MTA-R. For example: rejection by MTA-R of one among multiple destination addresses during the SMTP session.

The SAE layer.
The definition of an SAE layer, above the Internet email layer (see Figure 1), resolves these critical issues and introduces additional benefits.
We identify the sending SAE server with SAE-S, the receiving SAE server with SAE-R.
We define the format of the SAE-transport message, which is structured multipart MIME with one part containing the original email (message/rfc822 part) and another part containing various information about the email it transports, including the list of destination addresses (application/x-saeprotocol part).

SAE-S detects whether the domain contained in a destination email address is of SAE type, or is a normal Internet email address, through a system of certification of SAE domains that cannot be based solely on DNS, which is considered insecure. It is beyond the scope of this paper to define such a system.
From the same certification system it also obtains the name of the MTA-R that manages that domain.
It retrieves, again through the certification system, an email address on which the MTA-R provider wishes to receive SAE-transport messages.
It composes the SAE-transport message and sends it to that address.

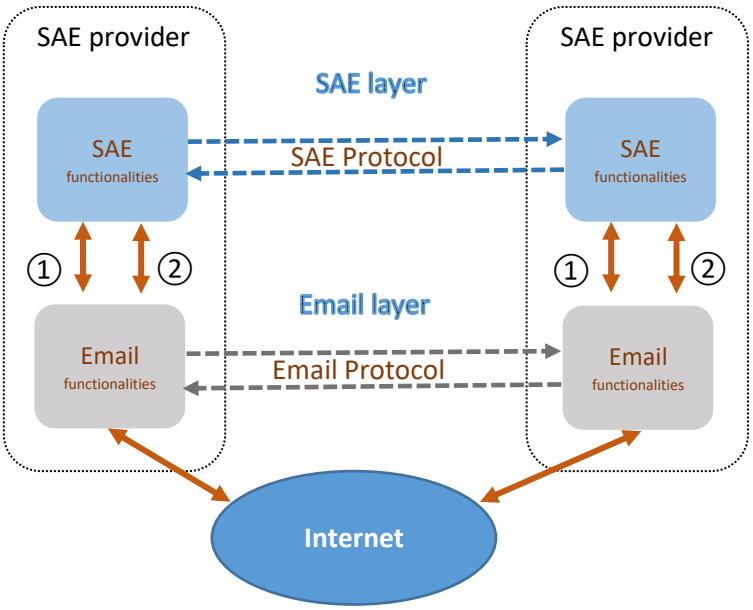
SAE-R by applying appropriate checks on both the IP address from which the SMTP session originates and the domain present in the sender email address, can determine whether the email came from an SAE provider.
It extracts the destination email addresses from the application/x-saeprotocol part that carries the SAE protocol data, extracts the original email from the message/rfc822 part, and delivers the email to the addresses extracted from the application/x-saeprotocol part. If required, it generates the corresponding delivery/non-delivery notification.
It can be noted that:
- the SAE-transport message is addressed to only one address of SAE-R even if the original message contained in the transport message has multiple destination addresses to SAE-R. In practice, the SAE-S (sender) address and the SAE-R (receiver) address are the interface addresses with which the underlying Internet email layer presents itself to the SAE layer (see layered architecture diagram in Figure 1). They represent the interfaces through which the SAE layer delivers/receives its protocol messages to/from the underlying Internet email layer (*data channel*).
- the addresses to which delivery occurs are actually those certified by the SAE-S, contained in the MIME application/x-saeprotocol portion of the SAE-transport message.
- the two CEM/email-Internet layers also interact through a control channel, over which, for example, the email-Internet layer can communicate any layer errors. Example: a SAE-transport message does not reach SAE-R because the Internet connection is down. The email-Internet layer sends a Delivery Status Notification to the interface address of SAE-S. SAE-S parses that notification to generate the necessary error report to the sender user of the original message.
- In composing the transport message, SAE-S must determine, for each address to which the original message is addressed, the destination SAE-R. In the event that a temporary error in finding the necessary information occurs for one or more of these addresses, SAE-S may send multiple transport messages for the same original message to the same SAE-R..
- the destination SAE addresses could be defined with any syntax, even different from the syntax defined by RFC 5322.

Encrypt of the transport message (SAE message).
SAE-S, during the information/data retrieval phase related to SAE-R, can also retrieve an X509 certificate, belonging to SAE-R, with which to do the encrypt of the transport message.
The encrypted transport message reaches SAE-R, which decrypts it with its own private key before delivering it to local recipients.
Advantages: protection from any malicious actions or configuration errors that direct the SAE message to a provider other than the actual destination provider.

Use of domain literal address format.
The SMTP protocol allows an IP address (example: mario.rossi@146.79.35.24) to be used instead of the domain of the email address.

The use of domain literal eliminates the query to DNS by which the SMTP client searches for the MX record; the SMTP session is opened directly with the IP address in the email address.
SAE-S, during the information/data retrieval phase of SAE-R, can obtain the interface email address of SAE-R in domain literal format.
SAE-S sends the transport message, composed as already seen, to that address.
The combination of the encrypt function of the transport message, with this define a security mechanism of the SAE layer, which is alternative to the corresponding Internet email layer (TLS).
The destination MTA server is "armored and trusted" with its IP address, which is contained in the SAE certification system. The problems of DNS insecurity (MX record) are bypassed. Even should the email reach the wrong server, it would be unreadable.
In practice, these security features of the SAE layer could allow TLS waiving of the SMTP session.
This feature of the SAE architecture could have interesting implications.

Overview of the SAE architecture.
Figure 3 shows the SAE layers and figure 4 shows the SAE architecture



① SAE protocol interface - ② control data interface
Figure 3 – Secure Application Email Layers

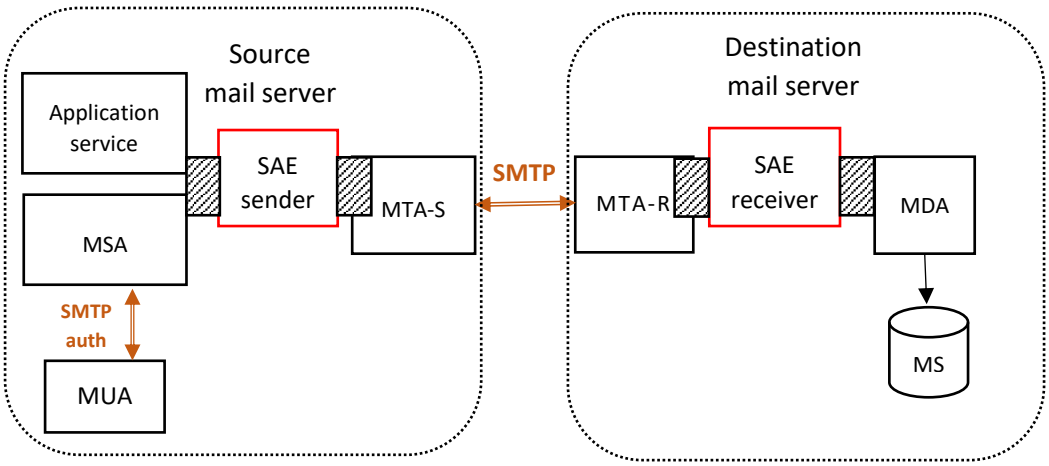


Figure 4 – Secure Application Email Architecture

From an architectural point of view we identify two SAE modules:

- SAE sender: receives the email message (original email) from the MSA or from a local application and creates and enqueues a SAE transport message to the MTA-S.
- SAE receiver: receives the SAE transport message from the MTA-R, extracts the original email from the SAE transport message and enqueues it to the MDA for the delivery to the user mailbox.

The SAE sender and the SAE receiver interfaces are identified by the SAE protocol interface addresses.
The SAE protocol interface address is an email address as defined by the RFC 5322 or it is an Address Literal as defined in 4.1.3 of RFC 5321:

- SAE data interface address: sae.data@destination-service-domain | sae.data@[destination-IP-address]
- SAE control data interface address: sae.control@destination-service-domain | sae.control@[destination-IP-address]

SAE message format.

SAE message are standard MIME [RFC 2045, RFC 2046] email messages.
There are two types of SAE messages:

- SAE transport message:
  It is formatted as MIME content-type multipart/mixed composed by two parts.
  - MIME content-type message/rfc822 that contains the original email as sent by the user.
  - MIME content-type: application/X-saeprotocol that contains the SAE control data.
- SAE control message: MIME content-type application/X-saeprotocol that contains the SAE control data.

Figure 5 and 6 show some examples of the SAE messages.

```
Date: Wed, 18 Jan 2023 10:22:41 +0100 (MET)
From: sae.data@[100.50.60.70]
To: sae.data@[10.20.30.40]
Subject: sae data message
X-SAE-transaction-id: abc123abc123@senderhostname
X-SAE-unique-message-id: zzxxcczzxxcc@senderhostaname
X-SAE-Version: 1.0

Content-type: multipart/mixed; boundary=--Boundary-AaBbCcDd

--Boundary-AaBbCcDd
Content-type: message/rfc822

Date: Wed, 18 Jan 2023 10:22:15 +0100 (MET)
From: alice@source.net
To: bob@dest.org, mark@dest.org
Subject: greetings
Content-type: text/plain

Hello,
blah blah ….

--Boundary-AaBbCcDd
Content-type: application/X-saeprotocol

Version: 1.0
SAE-transaction-id: abc123abc123@senderhostname
SAE-unique-message-id: zzxxcczzxxcc@senderhostaname
Snd-from: alice@source.net
Rcpt-to: bob@dest.org
Rcpt-to: mark@dest.org

… others attribute/value pairs for protocol extensions ….

--Boundary-AaBbCcDd--
```

Figure 5 – example of SAE transport message

```
Date: Wed, 19 Jan 2023 11:05:27 +0100 (MET)
From: sae.control@[100.50.60.70]
To: sae.control@receiverMXhostname
Subject: sae control message
X- SAE-transaction-id: SAE-transaction-id
Content-type: application/X-saeprotocol

Version: 1.0
Request-type: Domain
SAE-request-id: abc123abc123@senderhostanme
Rcpt-domain: dest.org

… others attribute/value pairs for protocol extensions ….
```

Figure 6 – example: SAE control request message

SAE request composition.
The SAE sender generates a unique SAE transaction id and composes a MIME email with Content-type: application/X-saeprotocol.
The body of the SAE request is composed by a list of attribute/value pairs as follows:

Request-type: *Domain*
SAE-request-id: *an id as per Message-id: from RFC 5322*
Rcpt-domain: domain of the *rcpt-to address from the envelope of the original email message*

SAE answer composition.
The SAE receiver compose a MIME email with Content-type: application/X-saeprotocol.
The SAE control answer is composed by a list of attribute/value pairs as follows:
Answer-type: Domain
SAE-request-id: *the request-id to which the answer is related*
SAE-answer-id: *an id as per Message-id: from RFC 5322*
SAE-domain: *the domain to which the following attributes are related*
SAE-certificate: *X509 certificate to encrypt the email*
SAE-domain: *the domain to which the following attributes are related*
SAE-certificate: *X509 certificate to encrypt the email*

Conclusion.

In this technical report we propose a new model relaying on the Internet email service: the Secure Application Email. The separation between the SAE layer and underlying Email Internet layer, allows the definition of the functionality and security level of an architecture, independent of the functionality and security of the underlying Email Internet layer. The two layers can evolve independently of each other. For example, currently the Internet Email layer could offer better security by adopting standards such as DANE/DNSSEC, but management difficulties encountered in adopting DNSSEC are delaying its deployment. This eventual evolution of the Email layer, of the authentication and reputation mechanisms of a message's sources, are entirely transparent to the SAE layer, which allows for additional features related to identification and security in email communications to be added.
The SAE layer allows the definition of verification and certification mechanisms, alternative to those currently defined for Internet Email, facilitating the deployment of solutions based on the SAE architecture regardless of the state of the underlying Email layer.