Requirements for Message Access Control
draft-freeman-plasma-requirements-08

Abstract

There are many situations where organizations want to protect information with robust access control, either for implementation of intellectual property right protections, enforcement of contractual confidentiality agreements or because of legal regulations. The Enhanced Security Services (ESS) for S/MIME defines an access control mechanism for email which is enforced by the recipient’s client after decryption of the message. The ESS mechanism therefore is dependent on the correct access policy configuration of every recipient’s client. This mechanism also provides full access to the data to all recipients prior to the access control check, which is considered to be inadequate for robust access control due to the difficulty in demonstrating policy compliance.

This document lays out the deficiencies of the current ESS security label, and presents requirements for a new model for providing access control to messages where the access check is performed prior to message content decryption. This new model also does not require policy configuration on the client thereby simplifying deployment and compliance verification.

The proposed model additionally provides a method where non-X.509 certificate credentials can be used for encryption/decryption of S/MIME messages.

The name Plasma was assigned to this effort as part of the IETF process. It is derived from PoLicy enhAnced Secure eMAil.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering
Table of Contents

1 Introduction ................................................. 4
  1.1 Data Access Control .................................. 4
  1.2 Encrypted E-Mail Using Web-based Credentials ........... 5
  1.3 Vocabulary ........................................... 6
  1.4 Keywords ............................................. 11
2 Background .................................................. 11
  2.1 ESS Security Labels .................................... 13
  2.2 Access Control and the Web .............................. 14
  2.3 Information Asset Protection ........................... 17
  2.4 Authentication Assurance Frameworks ....................... 18
  2.5 Electronic Signatures: Authentication vs. Authorization ... 18
3 Use Case Scenarios ......................................... 19
  3.1 Consumer to Consumer Secure Email ....................... 19
  3.2 Business to Consumer Secure Email ....................... 20
  3.3 Business to Business Ad-Hoc Email ....................... 23
  3.4 Business to Business Regulated Email ..................... 24
  3.5 Delegation of Access to Email .......................... 29
  3.6 Regulated Industry Email ................................ 29
  3.7 Email Compliance Verification ........................... 31
  3.8 Email Pipeline Inspection ............................... 31
  3.9 Distribution List Expansion ............................. 33
  3.10 Scalable Decision Making .............................. 33
  3.11 Related Use Case Scenarios ............................. 34
4 Plasma Data Centric Security Model .......................... 37
  4.1 Plasma Client/Server Key Exchange Level of Assurance ... 44
  4.2 Policy Data Binding .................................... 44
  4.3 Content Creation Workflow ............................... 46
  4.4 Content Consumption Workflow ............................ 48
  4.5 Plasma Proxy Servers ................................... 49
  4.6 Policy Types ........................................ 51
5 Message Protection Requirements ............................ 52
  5.1 General Requirements ................................... 52
  5.2 Basic Policy Requirements ............................... 54
  5.3 Advanced Policy Requirements ............................ 55
6 IANA Considerations ......................................... 57
7 Security Considerations ..................................... 58
Editorial Comments ............................................ 59
Appendix A References ......................................... 60
  A.1 Normative References .................................. 60
  A.2 Informative References ................................ 60
Appendix B Authors’ Addresses ................................ 61
Appendix C Document Change History ........................... 62
1 Introduction

The S/MIME (Secure/Multipurpose Internet Mail Extensions) standard [RFC5652] today provides digital signatures (for message integrity and data origin authentication) and encryption (for data confidentiality). The Enhanced Security Services (ESS) for S/MIME [RFC5035] provides for additional services including security labels (eSSSecurityLabel) which represent the access control policy. The label is a signed attribute in the signed data block of a message. The recipient of the message is responsible for checking that the recipient has a legitimate right to see the message based on the label. This type of security labeling is similar to that of stamping "Top Secret" on the cover of a document. It relies on the reader to not open and read the document when the policy is discovered.

The Cryptographic Message Syntax (CMS) [RFC5652] allows for a variety of different types of lock boxes to be applied to an encrypted message. This allows for a variety of security mechanisms to be used by the sender and the recipient to process the message. However the S/MIME standard is currently solely based on X.509[RFC5280] certificates. This means anyone without an X.509 certificate is unable to leverage the S/MIME protocol for securing email. The vast majority of users on the Internet have other forms of credentials (passwords, one time passwords, PGP keys etc.).

1.1 Data Access Control

There are many situations where organizations want to include information which is subject to regulatory or other complex access control policy in email. Regulated information requires some form of robust access control to protect the confidentiality of the information. While ESS for S/MIME [RFC5035] defines an access control mechanism for S/MIME (eSSSecurityLabel), it is an extremely weak form of access control as the recipient is responsible for the enforcement and is given access to the data even if the recipient fails to meet the access criteria as defined by the label.

An access control policy defines a set of criteria and evaluation logic that must be satisfied in order to grant access to the information.

* With Discretionary Access Control (DAC), policies are defined in terms of group membership the subject needs to meet the policy.

* With Role Based Access Control (RBAC), policies are defined in terms of what role the subject needs to belong to to meet the policy.
With Attribute Based Access Control (ABAC), policies are defined in terms of what attributes about the subject, their device or environment, their intended action on or use of the information and the resource are needed to meet the policy.

Examples of the types of attributes used in ABAC would include attributes about the subject such as their employers identity, their nationality, citizenship etc., or attributes about their device such as its name, boot state. Standards now exist that enable the transport of attributes [SAML-overview].

An ESS Security label is a signed attribute of a SignedData object which indicates the access control policy for the message. While an ESS Security Label provides a standardized representation of an access policy identifier, it does not define any methods of obtaining the necessary information to satisfy the policy or policy description in order to enforce the policy. The fact that this is a signed attribute protects the integrity of the ESS label and provides a tamper evident binding of the label to the message but does not by itself protect the confidentiality of the message. At the point where the recipient learns the access control policy to enforce on the data the recipient already has access to the data. While the signature provides tamper evident integrity for the label over the clear text, it is not tamper proof because it is susceptible to unauthorized removal if the message only contains a SignedData element, i.e. any Message Transport Agent (MTA) in the path can remove a signature layer of a SignedData message thereby altering the access control data. Encrypting the signed message protects the confidentiality of the data and protects the SignedData from tampering from anyone unable to decrypt the message. However, encrypting the message means that no intermediate agent can enforce the label policy and it does not protect the label from any entity that has the ability to decrypt the message.

From a regulatory enforcement perspective, ESS labels are an extremely weak form of access control because cryptographic access to the data is given before the access check. The correct enforcement of the access check is dependent on the configuration of every recipient’s email client. Since the cryptographic access is granted before the access policy check, there is no cryptographic impediment for a recipient who is able to decrypt the data but is unauthorized under the policy, to ignore the policy and access the data. A stronger enforcement model is needed for regulatory control for email where cryptographic access is only granted after the access check is successful.

1.2 Encrypted E-Mail Using Web-based Credentials
There are many users on the Internet today who have forms of authentication credentials other than X.509 certificates. S/MIME today can only use X.509 certificates to protect the confidentiality or the data origin authentication of the messages. This means the many users without X.509 certificates cannot use S/MIME. Standards-based services (e.g. [SAML-overview]) are now available which abstract the specifics of an authentication technology used to identify a subject from the application itself (S/MIME in this case). Adoption of this abstraction model would enable a broader set of authentication technologies to be able to use S/MIME to secure email for confidentiality or data origin authentication. It also allows for new authentication technology to be deployed without impacting the core protocol.

1.3 Vocabulary

Many of the terms used in this document are the same as as defined elsewhere [RFC3198]. Some terms, though representing broadly the same concept have minor differences than defined elsewhere so the following clarifies their use in this document.

**Attribute Based Access Control (ABAC)** Where the access control policy is specified by a set of attributes, their values, and any relationship between attributes required to authorize an action on a resource. These attributes may be provided by the subject as part of the decision request (Front End Attribute Exchange) or discovered by the policy decision service itself (Back End Attribute Exchange). The policy, for example, may require attributes about the subject, their device or environment, a resource, or the intended use of the information.

**Back End Attribute Exchange (BAE)** When subject attributes are directly sent from the Policy Information Point (PIP) to the Policy Decision and Enforcement Point (PDEP) i.e. they are not relayed via the Decision Requestor (DR).

**Capability Based Access Control (CBAC)** Where access control is via a communicable, unforgeable token. A capability token is a protected object which, by virtue of its possession by a subject, grants that subject the capability.

**Cipher text** Plain text which has been processed by an encryption algorithm to render it unreadable by
a program or human without the appropriate cryptographic key.

Confidential
A message has been protected to a known level of confidence so that the contents are not decipherable by unauthorized users.

Content Encryption Key (CEK)
A key used to encrypt protected end user data. (See Key Encryption Key)

Cryptographic Lock Box
A data structure which holds a CEK encrypted for a specific user. CMS implements Cryptographic Lock Boxes as RecipientInfo structures.

Data Origin Authentication
Enables the recipient to verify that message has not been tampered with in transit or in storage, and the subject who originated the message.

Decision Requester (DR)
The service responsible for making policy decision requests to the PDEP. In this model the policy decision is enforced by the PDEP by its control of cryptographic keys. The DR enforces any obligations the PDEP may require such as signing or encryption of the data, generating audit events etc. A DR is distinct from a PEP in other models such as XACML in that a DR is not by default trusted with the clear text data. Policy enforcement is performed by the PDEP. A DR may establish trust by presentation of attributes about itself and its environment to show it is trustworthy.

Early Binding
The concept of creating the cryptographic lock box for a recipient at the time the message is sent. (Compare with Late Binding)

Front End Attribute Exchange (FEE)
When subject attributes are relayed by the DR from the PIP to the PDEP i.e. they are not sent directly.

Integrity Protected
A recipient of a message can determine to a known level of confidence that a message has not been modified between the time that it was created and when it was received by the recipient.

Key Encryption Key
A key used to encrypt another cryptographic
(KEK) key, often a CEK. (See Content Encryption Key)

Late Binding The concept of creating the cryptographic lock box for a recipient when the recipient attempts to decrypt the message. Late binding has a potential downside because the sender cannot know what symmetric algorithms the recipient supports which can lead to interoperability issues. (See Early Binding)

Level of Assurance (LoA) A quality grade assigned following the completion of a security evaluation. For example, it can be used for an Identity where it provides the quality of the identity of a subject. It can also be used to represent the quality of a products or services Common criteria evaluation.

Mail Transfer Agent (MTA) A program that transfers email from one computer to another. An MTA implements both the sending and receiving of email.

Mail User Agent (MUA) A program or service used to manage a user’s email. The MUA may be a program run on the users computer or a Web service accessed via the users browser.

Metadata Metadata is data about data. There are three kinds of metadata.

(1) Content metadata is metadata about an instance of data, the actual data content. An example of content metadata would be "this data contains Company Foo intellectual Property" or "this is a patient record".

(2) Policy metadata is metadata about the policies to apply to an instance of data. An example of policy metadata would be "apply Company Foo XYZ policy".

(3) Structural metadata is metadata about the design and specification of the data. An example of structural metadata would be "this is a patient record table".

Orthonym The correct or legal name of a place, person or thing. (See Pseudonym.)
### Plain text

The information in a state which can be directly read by a human or an appropriate application.

### Policy

1. A statement in a human language that defines a course of action by an individual or organization. These statements may be in the form of legislation, regulation, a legal contract, or organization goals.
2. Technical controls for implementation of the human-readable policies. Policies may stipulate many forms of technical controls requirements such as data protection, access control, data integrity, data origination authentication, data retention, etc.

### Policy Administration Point (PAP)

The system entity that creates, maintains, and publishes policies or policy collections. The policies define the rules, their conditions, and actions associated with the policy.

### Policy Collection

A collection of one or more policies which is associated with a role. The policy collection may also define the logical relationship between the policies. Each collection is identified by a name known as a role name.

### Policy Decision Point (PDP)

The system entity that evaluates the policy using attributes supplied by a PIP to render decisions on requests made by PEPs. The PDP does not enforce the decision.

### Policy Decision and Enforcement Point (PDEP)

The system entity that evaluates the policy criteria published by a PAP, using attributes supplied by a PIP to render decisions on requests made by DRs. The PDEP is able to enforce its decision via the use of cryptographic keys.

### Policy Enforcement Point (PEP)

The service responsible for making policy decision requests to the PDP. A PEP has access to the plain text data and metadata and enforces the decisions made by the PDP.

### Policy Identifier

The tag that is used to identify a policy. For the purposes of this document the focus is on two different types of policy identifiers.
Object Identifiers (OIDs) are what are currently used in many security policy systems and are the only method of policy identification supported by ESS security labels. Additionally URIs are supported as policy identifiers as they provide a more user-friendly method to uniquely identify a policy and allow discovery of the policy.

<table>
<thead>
<tr>
<th>Policy Information</th>
<th>A service which issues assertions, for example about a subject, their device, or environment e.g., a SAML Security Token Service. This model supports both front end and back end exchange of assertions between the PIP and the PDEP. Attributes can be distributed directly between the PIP and the PDEP (in the case of BAE). Alternatively attributes may be distributed via the DR (in the case of FAE). There are two types of PIP based on the types of attribute the PIP would assert about the subject. An Identity Provider (IdP) PIP will issue authentication attributes e.g., information about how and when the subject authenticated to the IdP. An IdP may also issue attributes about the subject themselves, e.g., their full name, age, or citizenship. An attribute provider (AtP) PIP only issues attributes about the subject or the subject’s environment.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy Label</td>
<td>The data structure which holds one or more policy identifiers and their logical relationship.</td>
</tr>
<tr>
<td>Pseudonym</td>
<td>A name that a person or group assumes for a particular purpose, which differs from their original or true name. (See Orthonym.)</td>
</tr>
<tr>
<td>Role</td>
<td>An abstract subject which has a series of authorizations assigned to it. Users are assigned to roles to perform the duties of the role. Users typically select a role to perform a function to disambiguate which role they are currently functioning as. A role is distinct from a group because a group is a collection of subjects which has no intrinsic authorizations.</td>
</tr>
<tr>
<td>Role Based Access Control (RBAC)</td>
<td>Access control based on the assignment of a role. Subjects are then allowed to assume</td>
</tr>
</tbody>
</table>
one or more roles based on their job needs, for as long as their job requires.

Role Token

A token issued to a subject containing one or more Policy Collections. The role token is used as part of policy discovery and management in Plasma. It is not used as part of access control decisions in any way.

We additionally make use of the following terms:

- **Attribute**
  The act of a client requesting and obtaining a set of attributes for a subject. The issuance of attributes will itself be controlled by policy and thus recursively embeds this same picture in that process. The attributes can be requested either by the DR (if using FAE) or the PDEP (if using BAE).

- **Content Protection**
  The protocol to be run by the DR to get the set of decisions and information required to successfully create and encode a data block with appropriate labeling. This protocol is part of the work to be defined by this group.

- **Content Consumption**
  The protocol run by a DR to obtain the permissions and information needed to decode and access data with appropriate labeling.

- **Content Distribution**
  Can be any of a number of methods by which the content is transmitted from the Content Creator to the Content Consumer. These methods include, but are limited to: email, FTP, XMPP, HTTP and physical distribution.

1.4 Keywords

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

2 Background

The S/MIME standard [RFC5751] provides a method to send and receive secure MIME messages. S/MIME uses CMS[RFC5652] as the means to protect the message. While CMS allows for many types of security credentials to be used, S/MIME [RFC5750] exclusively uses X.509
certificates [RFC5280] for the security credentials for signing and encryption operations. S/MIME uses an early binding mechanism for encryption keys where the sender needs to discover the public key for each recipient of an encrypted message before it can be sent. This requires the sender to maintain a cache of all potential recipient certificates (e.g. in a personal address book) and/or have the ability to find an acceptable certificate for every recipient from a repository at message creation. This key management model has limited the use of S/MIME for encryption for a variety of reasons. For example:

- The recipient may not have an X.509 encryption certificate
- The sender may not have received a signed email with the recipient’s certificate
- The recipient may not have an available repository from which to retrieve the recipient’s certificate
- The sender may be unaware of the location of the recipient’s repository
- The recipient’s repository may not be accessible to the sender, e.g., it’s behind a firewall
- The sender may not have a valid certificate path to a trust anchor for the recipient’s certificate

If one or more recipient certificates are missing, then the sender is left with a stark choice: send the message unencrypted or remove the recipients without certificates from the message.

The use of secure mailing lists has the ability to provide some relief to the problem. The original sender does not need to know the appropriate encryption information for all of the recipients of the mailing list, just for the mailing list itself. It can thus be thought of as a form of late-binding of recipient information for the originating sender. However it is still early-binding encryption for the mail list agent; as it needs to perform all of the gathering and processing of certificate information for every recipient that the agent will relay the message to.

In many regulated environments end-to-end confidentiality between sender and recipients by itself is not enough. The regulatory policy requires some form of access control check before access to the data is granted. In many inter-organization collaboration scenarios it’s impossible for the sender to satisfy the access checks on behalf of all recipients since they don’t have, and frequently should not have
access to, all the recipient’s attributes because to do so may be a
breach of the recipient’s privacy. Indeed to release the attributes to
the sender may require that the sender’s attributes first be released
to the recipient’s attributes provider. It’s a fundamental tenet of
good security practice that users should control the release of data
about themselves.

2.1. ESS Security Labels

Security labels are an optional security service for S/MIME defined
in Enhanced Security Services for S/MIME [RFC5035]. The ESS security
label allows classification of the sensitivity of the message
contents using a hierarchical taxonomy in terms of the impact of
unauthorized disclosure of the information [RFC3114]. The security
label can also indicate access control such as full time employees
only or US nationals only. ESS security labels are authenticated
attributes of a CMS signer-info structure in a SignedData object.
The label when applied to signed clear text data provides the access
control decisions for the plain text. If applied to cipher text such
as the outer layer of a triple wrapped S/MIME message the label is
used for coarse grained optimization such as routing.

2.1.1. Problems With ESS Security Labels

ESS Security Labels have been found to have a number of limitations.

1. When the label is on the innermost content, access to the plain
text is provided to the recipient (in some form) independent of
the label evaluation as it will be processed for the purpose of
hash computation as part of signature validation. Depending on
how a triple wrapped message is processed by the recipient’s CMS
code, the inner content may be processed for signature validation
even before the outer signature is validated. This would happen
for a stream based CMS processor which starts processing inner-
layers immediately rather than finishing processing of each layer
and caching the intermediate results.

2. Labels applied can be removed in transit. If a signed layer is
seen then it can be removed by any agent that processes the
message (such as a Message Transfer Agent). If the label is
protected by an encryption layer then it can only be removed by
any agent that has a decryption key (Encryption Mail List agents
or Spam Filtering software would be two such examples).

3. Policies are identified by Object Identifiers. This makes for a
small tight encoding, but it does not provide any mechanism for
an email client to discover how to enforce a new access control
policy if the message contains a policy the client is unaware of.
This provides an impossible choice: ignore the access control policy and grant access to the message or block access to the message. Object identifiers also do not provide a good display name for a user so that they could manually find and download a new policy.

4. The current ESS standard only allows for a single policy label in a message; no standardized method of composing multiple policy labels together has been defined. This is adequate for coarse-grained policy binding to express a limited set of choices such as with information sensitivity which typically provides a hierarchy of 3-5 choices. Many data sets need to be subject to multiple access control policies. For instance, a message may contain information that is both propriety and export controlled. Trying to represent combinations of policies via a single policy label would lead to an exponential growth in the number of policy labels.

5. ESS Labels do not provide for any auditing of who has been granted access to the message. All policy evaluation is local to the recipient’s machine; no centralized logging of access to the message can be performed.

6. Enforcement of the policy occurs on the recipient’s machine; the compliance with the policy is dependent on the state of the configuration of every receiving agent. The policy is enforce by whatever module is located on the user’s system. For cross corporate systems, this means that the policy provided by Company A must be installed on Company B machines, or Company B must install a policy that Company A will accept as being equivalent to their own policy. Additionally, any time that a new version of the policy module is rolled out, there will be a time lag before every recipients machine will have the updated module. This makes policy compliance practically impossible in anything but a small, closed environment.

7. Access to the message cannot be granted or removed after the message has been sent. Therefore, if a recipient has a designated alternate recipient they will not be able to read the message. Also, if the sender subsequently learns one of the recipients was in error, they cannot correct the mistake.

2.2. Access Control and the Web

A prerequisite for many Web transactions is the disclosure of attributes about the subject such as name, age, email address, physical location, address, credit card number, social security number, etc. Some attributes lend themselves to easy verification but
many do not. An assertion of an email address can be verified by sending an email to the address containing a secret ephemeral challenge. Subsequent demonstration of knowledge of the ephemeral challenge verifies the email address assertion. Other assertions such as "this is my credit card account number" are not easily verified. The fact that it is a valid credit card number can be verified but not the binding to the subject attempting to use it. Where a claim is not easily verified it is often combined with other assertions under the assumption that knowledge of this larger data set verifies all the assertions in the data set. If you know the account number, billing address, etc., ‘of course’ you must be the account holder. This is a very weak form of verification as is often demonstrated by the growth of identity theft; much of this bigger data set is often publicly available via social networks or easily guessed, e.g., the most popular professions for a parent is dead or retired. Many of the assertions which are harder to verify are based on government issued documents such as a birth certificate, driver’s license, identity card, or passport. This requires an exchange of the documents between the relying party and the subject. For a small number of high value transactions (e.g., opening a new account) with relying parties that have widespread physical presence (e.g., a bank or Post Office) this is acceptable because the applicant can present themselves with the required documentation in person. However, Web-based relying parties cannot perform an in person exchange of documents to verify information on government issued documents. The approach taken with such relying parties is to have trusted assertion providers where the assertion provider can perform an in-person exchange of documents with the subject then vouch for the set of assertions they have verified.

SAML [SAML-core] defines an XML framework for describing and exchanging assertion tokens containing attributes about subjects. The entity issuing the assertion tokens about the subject is known as the assertion provider. The entity consuming the assertion with the attributes is known as the relying party. The well-known scenarios for using SAML are:

- Single Sign On across systems on different platform technology
- Federated Identity between business partners
- Web Services and other standards, e.g., SOAP based protocols

X.509 identity certificates are signed tokens that can exchange a limited set of identity attributes about a subject, such as an X.500 distinguished name, email address, Kerberos principal name, etc. X.509 identity certificates are issued to a subject and contain the
subject’s name and a public key. The public key provides a replying party a mean to authenticate the subject’s identity either via a challenge by to the subjects matching private key or by the subject to prove data origination authentication by signing data with the private key. The X.509 identity certificate can be supplemented by an X.509 attribute certificate with can contain additional attributes about the subject such as date of birth, business sign-off limit levels, etc.

SAML also uses signed tokens and supports an extensible set of attributes about the subject as is supported by X.509 identity and attribute certificates. SAML tokens can be either Bearer Tokens or Holder-of-Key tokens. Bearer tokens have no cryptographic key and their security is based on the time between when the token was issued and time it was presented to the replying party together with the token being issued for use with the replying party. Low value transactions can use Bearer tokens where possession of the token alone is considered acceptable for the transaction risk. Holder-of-Key tokens contain a cryptographic key (either public or symmetric) and like X.509 identity certificate the subject proves their identity to the replying party by demonstrating control over the key e.g. signature or HMAC over some data. Higher value transactions can therefor have a stronger proof of identity by the demonstration of proof of possession of cryptographic keys. SAML also defines a number of query/response style profiles for issuing the tokens enabling both client and relying parties to request SAML tokens of different types.

The critical difference between SAML and X.509 identity certificates is that SAML [SAML-QUERY] also abstracts the details of the authentication protocol from the relying party. The attribute provider can use a broad range of authentication mechanisms such as passwords, one-time passwords, biometrics, X.509 certificates, etc., to authenticate the subject without impacting the relying party as part of the token issuing process. The attribute provider includes the details of the authentication mechanism used or its strength using an established strength of authentication scale such as NIST SP800-63-1 [SP800-63-1]. The relying party then inspects the attributes in the token from the identity provider to determine if they complies with its access policy.

2.2.1 Attribute Exchange Models

Access control in distributed environments depends on the creating, communicating of assertions with attributes about subjects. The model has asserting providers issuing attributes about subjects and relying parties consuming attributes about subjects. A subject can be a human, a device or service. The subject must have a relationship with
the assertion provider since they have been through some form of registration process with the provider. There is no requirement to have a relationship between the assertion provider and a relying party. The relying party must trust the assertion provider, but not vice versa. This is the same model as exists with X.509. The subject must have a relationship with the CA, the relying party must trust the certificates issued by the CA, but there is no requirement for the CA to have any form of relationship or trust with the relying party. Release of subject attributes to a relying party must be under a policy due to the sensitivity of the data. The subject’s themselves can request and give approval for the release of attributes from the assertion provider and relay them to the relaying party (Front End Attribute Exchange). If the subject has given prior consent, the relying party may receive attributes directly from the assertion provider (Back end Attribute Exchange).

2.3. Information Asset Protection

Information Asset Protection (IAP) is a concept developed by the Transglobal Secure Collaboration Program (TSCP), a working group comprised of the major players in the western Aerospace and Defense industry. The industry is highly regulated and operates in an environment with many policies governing the access to information assets. These policies are motivated by the desire to protect intellectual property, the confidentiality of information, or are imposed by government regulations such as the US International Traffic in Arms Regulations (ITAR) from the US Department of State. They apply to the information assets in whatever form the asset may take and are independent of the application used to create the information. These policies take many forms, e.g., verification the recipient has demonstrated a need to know the information because they are working on a specific project, that they have passed the appropriate background and nationality checks, or that they have signed the appropriate non-disclosure agreement. What is needed is a policy driven, information-centric protection where the applicable policies either is manually or automatically attached to the information, and based on the policy, the system understands what access control and data protection is necessary.

Email is an application widely used in the Aerospace and Defense industry. S/MIME is widely used today and provides sender to recipient confidentiality. This protects the contents of the message from discloser to unauthorized third parties, e.g., while it is in transit between MTA’s or while at rest in an MTA message queue or recipient’s mailbox. However, it does not impose any finer-grained access control such as those required by many policies. S/MIME does define an extension mechanism for access control via an ESS security label [RFC5035] though this mechanism has drawbacks (see above).
2.4. Authentication Assurance Frameworks

A number of organizations have created taxonomies to define the possible levels of identity assurance for electronic authentication. The objective of the framework is to provide a simple abstraction of the details of:

- Identity proofing and registration of subjects
- Tokens used by subjects for providing electronic identity
- The token management mechanisms
- Protocols used for subject to employ tokens to authenticate to an identity provider
- Protocols used by subjects to authenticate and pass attributes to a relying party

These frameworks have been drafted by industry organizations [lib-iaf][kan-iaf] and governments [SP800-63-1]. While all of these frameworks may not agree on every aspect, at a macro level they do exhibit many similarities. A common theme in many is the adoption of a small number of levels of identity assurance, typically between 3 and 5. A simplified description of the levels is:

- **Level 1** Negligible confidence in the asserted identity
- **Level 2** Some confidence in the asserted identity
- **Level 3** Significant confidence in the asserted identity
- **Level 4** High confidence in the asserted identity

A framework defines broad characteristics in the area of identity proofing, credential type and management, identity provider authentication and relying party authentication.

2.5 Electronic Signatures: Authentication vs. Authorization

Electronic signatures on email are used today to show data origination so only authentication is required. However, with transactions that are legally or regulatory significant, authentication alone is frequently insufficient. Policy requires other factors to be considered to ensure that the transaction meets policy requirements. Examples of these factors include:

- The state of the system generating the signature
o An indication of the signer’s intent

o Attributes about the signer to indicate, for example, job
  function in the company, job assignments, professional
  qualifications, signing authority, etc.

Many organizations would like email based work flows to be an option for these transactions.

3. Use Case Scenarios

This section documents some email based use cases that the new protocol aims to support. Also included are some related scenarios where the same underlying theme of consistent policy enforcement equally applies.

3.1 Consumer to Consumer Secure Email

One of the issues that is stopping the use of secure email in personal mail is the fact that consumers find X.509 certificates difficult and expensive to obtain and then use - especially across a set of devices (phone, tablet, workstation). One of the possible use cases of Plasma is to try and deal with this by removing the dependency on X.509 certificates. The details of the use case are therefore: Alice wants to send an email message to Bob that contains sensitive, personal data so she is concerned about ensuring only Bob can read it. Bob has a strong credential he can use to identify himself, but it’s not an X.509 certificate. Alice needs to ensure the following:

(a) Only Bob can read the email.

(b) Bob has the ability to verify the email is from Alice.

(c) Bob has the ability to verify the email message has not been modified since Alice sent it.

The sequence of events could be as follows:

1. Alice composes the email to Bob.

2. Alice’s email client allows her to classify the email. Alice classifies the email as Personal Communication which is a policy provided by her ISP.

3. Alice’s email client knows the protections to apply to a Personal
Communication; it knows to encrypt and sign the message.

4. The protected email is able to flow securely and seamlessly through existing email infrastructure to Bob. The data is protected while in transit and rest.

5. Bob receives the email and sees that it is a secure message. Bob can verify that the secure message has not been altered. Bob attempts to open and decrypt the email. If Bob is on the same ISP as Alice, then the same username/password as he uses to get his Email is used to obtain the needed keys. If Bob is on an ISP that is federated with Alice’s ISP then an infrastructure such as SAML, OpenID, OAUTH or ABFAB could be used to validate Bob’s identity and allow the needed decryption keys to be released.

3.2. Business to Consumer Secure Email

There are many examples of business-to-consumer secure email scenarios where the email could potentially contain sensitive medical or financial data. This would include doctor-patient; bank-account holder; Medical insurance-insured person and mortgage broker-customer communications. Two examples are presented here.

3.2.1 Bank Statement Email

A bank (The Bank of Foo) has determined that it will be using email to distribute statements to its customers (Bob). The information is confidential, so any channel of communication the bank selects must protect Bob’s privacy. The bank needs to ensure the following:

(a) Only Bob (or additional owners of the account) can read the email

(b) Bob authenticates with a sufficient level of identity assurance. The same identity assurance authentication level used to do online banking would be considered sufficient

(c) Bob can verify the statement is from his bank

(d) Bob can verify the statement has not been modified since his bank sent it.

The sequence of events would be as follows:

1. As part of routine end-of-the-month processing, the bank composes an email to Bob. They include the statement of balances and activity either as an attachment or as the body of the message.
2. The statement mailer for the Bank of Foo has been configured to apply a specific policy to the email.

3. The statement mailer for the Bank of Foo knows the protections to apply based on the policy; it knows to encrypt and integrity protect the message and what level of assurance is required for the recipients identity.

4. The protected email is able to flow securely and seamlessly through existing email infrastructure to Bob. The data is protected while in transit and at rest.

5. Bob receives the email and sees it is a secure message from the Bank of Foo. Bob can verify the message has not been altered as it is signed by his bank. Bob uses the same credential as he would for online banking to prove his identity to the email system and obtain the keys necessary to decrypt the message.

The same process could be used for any messages sent between the bank and its customers. Thus, messages dealing with loan applications and changes in bank policies can be sent out in the same manner, potentially using different policies. In some of these cases it might be in the bank’s interests to record in an audit trail if and when the keys were handed out on certain emails. For a statement, the bank would not expect a reply to occur, however, for other types of messages it should be possible for Bob to reply under the same level of protection. If Bob is able to use the same credential when sending a message, as the one he uses for accessing the bank’s Web site then the bank has the same assurance of the message sender’s identity.

3.2.2 Doctor-Patient Communications

In the second example, let’s say that Alice is a doctor and has received test results for her patient Bob. This information is confidential and regulated, so any channel of communication she selects must protect Bob’s privacy and comply with regulatory requirements. Alice elects to use email to reach Bob quickly and easily with news of the results. In this respect it is similar to the previous use case; however there are some additional complications that might need to be dealt with as well. Depending on who Bob is and where he is currently, there are additional people that may also need to be automatically informed of the same information, or need to have the ability to access the contents of the message. Examples of these would be Bob’s spouse, an individual who is making care decisions for Bob (i.e., one of Bob’s parents), or an individual in charge of dealing with Bob’s day-to-day health care (i.e., a charge nurse in a hospital or a visiting nurse). All of
these people may have the same need-to-know as Bob. There is also the possibility that some parts of the message may need to be released to some individuals but not to others. As an example, the mail message could contain a prescription; that specific portion of the message may need to be read by Bob’s pharmacist. Alice needs to ensure the following:

(a) Recipients can read the parts of the email they are authorized to see. The definition of authorized will vary with the content of the message and thus the policy applied. (For example, general health issues will certainly be treated differently than mental health issues, even by a General Practitioner.)

(b) That Bob is required to authenticate with an identity assurance level 2 or above.

(c) That Bob can verify the email is from Alice.

(d) That Bob can verify the email has not been modified after Alice sent it.

The sequence of events would be as follows:

1. Alice composes the email to Bob. She includes some comments and suggestions for Bob and attaches the test results.

2. Alice’s email client allows her to classify the email. Alice classifies the email as a Doctor-Patient communication. As a side effect of classifying the email message, the policy may suggest or mandate additional individuals that the communication should be addressed to.

3. Alice’s email client knows the protections to apply to Doctor-Patient communication; it knows to encrypt and integrity protect the message.

4. The protected email is able to flow securely and seamlessly through existing email infrastructure to Bob. The data is protected while in transit and at rest.

5. Bob receives the email and sees it is a secure message from Alice. Bob can verify the message has not been altered. Bob attempts to opens the email. Bob provides a Level 2 password to retrieve the necessary decryption keys. After Bob has proved his identity, he is able to read the email.

There are number of different places where the identity provider for Bob could live. The first is at Alice’s office; Bob already has a
face-to-face relationship with Alice and the credential could be setup in her office. A second could be Bob’s insurance provider. Bob has a relationship with his insurance provider as does Alice, thus it can serve as an trusted identity provider to healthcare providers. A third location could be a federation of doctors in an area, potentially with other health providers (such as hospitals and convalescent centers). Bob has setup an identity with Alice, but if he gets referred to Charlie by Alice for some procedures, Charlie would not need to setup a new identity for Bob but instead could just refer to Alice for the necessary identity proof. Many of these types of situations are dealt with by [I-D.ietf-abfab-arch].

There are a number of other additional services that could be provided by the policy system. One example would be that if the information was time critical, if Bob does not access his message within a given time period, the policy server could notify Alice of this fact so that an alternate method of communication can be attempted with the same information.

3.3 Business to Business Ad-Hoc Email

Early in the relationship between two companies, it is frequently necessary to exchange sensitive information as a preliminary to a more formal business relationship e.g., contract negotiations. This needs to occur before the relationship has matured to the point that a formal relationship is reflected through a specific legal agreement. Business owners need the agility to interact with potential partners without having to engage their respective IT staffs as a prerequisite of the communication.

As an example, Charlie works for Company Foo. He has just met Dave from Company Bar to discuss the prospect of a potential new business opportunity. Following the meeting, Charlie wants to send Dave some sensitive information relating to the new business opportunity. When Charlie sends the email to Dave with the sensitive content, he must ensure the following objectives:

(a) Only Dave can read the email

(b) Dave is required to authenticate with an identity assurance level 2 or above

(c) That Dave can verify the email is from Charlie

(d) That Dave can verify the email has not been tampered with

(e) Charlie may also need to keep a record of the fact that Dave accessed the message and when it was done.
The sequence of events Charlie would use is as follows:

1. Charlie composes the email to Dave. He includes some sensitive information relating to potential terms and conditions for the new contract that Foo and Bar would sign to form a partnership for the business opportunity.

2. Charlie’s email client allows him to classify the email. He classifies the email as an ad-hoc pre-contractual communication.

3. Charlie’s client knows the protections to apply to ad-hoc pre-contractual communication; it knows to encrypt and integrity-protect the message and the level of assurance required for the recipient’s identity.

4. The protected email is able to flow securely and seamlessly through existing email infrastructure to the recipient (Dave in this case). The data is protected while in transit and at rest.

5. Dave receives the email and sees it is a secure message from Charlie. (Charlie policy requires level 2 authentication for which, Dave uses a password). Dave is able to prove his identity to the level of assurance requested by Charlie so he is able to read the email. The organization Dave works for has an identity service which he uses to prove his identity for Charlie’s email. Dave opens the email.

If Dave or his delegate replies to the email from Charlie, the new message inherits the policy from the original messages so the entire message thread has the same policy. The policy also applies to messages forwarded by Dave because it contains information from Charlie and Company Foo wants consistent policy enforcement on its information.

3.4 Business to Business Regulated Email

As business relationships mature they often result in a formal contractual agreement to work together. Contractual agreements would define a number of work areas and deliverables. These deliverables may be subject to multiple corporate and/or regulatory policies for access control, authentication and integrity. Some classes of email may have information which is legally binding or the sender needs to demonstrate authorization to send some types of message where authority to send the message is derived from their role or function. Also many regulated environments need to be able to verify the information for an extended period – well beyond the typical lifetime of a user’s certificate. The set of policies applicable to an email is potentially subject to change as the different user’s contribute.
information to the email thread.

### 3.4.1 Regulated Email Requiring a Confidentiality Policy

Company Foo has been awarded a contract to build some equipment (Program X). The equipment is covered by export control which requires information only be released to authorized recipients under the terms of the export control license. Company Bar is a foreign subcontractor to Company Foo working on Program X. Company Foo sets up some business rules for access to program X data to ensure compliance with the export control license requirements. Company Foo also set up separate rules to cover the confidentiality of its intellectual property contributed to Program X. Company Bar also sets up its own policies to protect the confidentiality of its own intellectual property it contributes to Program X. As part of the agreement between Foo and Bar, they have agreed to mutually respect each other’s policies.

Confidentiality policies can change over time. It is important to be able to implement the changes without the need to update the data itself to reflect the change as finding all instances of the data in an intrinsically impossible problem to solve.

Frank is an employee of Company Foo. He has been assigned as a design team leader on Program X and as an individual contributor on Program X integration. Frank wants to send some email as a team leader to colleagues working on Program X in both Companies Foo and Bar.

Grace is an employee of Company Bar. She has also been assigned to the design team of Program X.

When Frank sends the email with Program X regulated content he must ensure compliance with the export control policies. When Frank sends a Program X email he must ensure recipients are authorized to read the contents to ensure Company Foo remains in compliance with its export control license.

If Frank also includes Company Foo intellectual property in an email, he must also ensure recipients are authorized to read the intellectual property contents.

When Grace receives a Program X email, she must provide attributes about herself to prove compliance with the export control policy. If the email also contains Company Foo intellectual property, she must also provide attributes to show she is authorized to read the information under the agreement between Company Foo and Company Bar.
If Grace sends an email with Company Bar intellectual property, she must ensure recipients are authorized to read the contents under the agreement between Company Bar and Company Foo.

When Frank sends a Program X email he must ensure the following objectives:

(a) Only recipients who meet the Program X export control policy and/or Company Foo’s intellectual property protection policy can read the email.
(b) Recipients authenticate with a identity assurance level 3 or above.
(c) Recipients present all other attributes about themselves necessary to verify compliance with the applicable policies (their program assignment, nationality, professional or industry certifications, etc.).
(d) Recipients can verify the email is from Frank to the level of identity assurance as defined by the message policy (i.e., level 3 or above).
(e) Recipients can verify the email has not been tampered with to the level of identity assurance as defined by the message policy.
(f) Recipients are made aware that the message is a Program X email (and the contents can only be shared with other Program X workers) and/or the message contains Company Foo’s intellectual property.

The sequence of events Frank would use is as follows:

(1) Frank composes the email and includes a Program X distribution list as a recipient. He include some information related to Program X. Frank also includes some information which is Company Foo’s Intellectual Property.
(2) Frank’s email client allows him to select the Program X role. The client then allows Frank to select from a set of policies appropriate for Program X.
(3) Frank selects the Program X content and Company Foo IP policies from the list of available policies.
(4) The email client knows to encrypt the message, the key size and algorithm to use. It also knows that the message needs to be signed with a level 3 or above private key.
(5) Frank clicks the "send email" button. The client signs the email using his smart card private key and includes the certificate with the appropriate public key for verification of the signature by recipients. The Client then encrypts the message and obtains data from a server that will enforce the access
control requirements for Frank, and sends it to his email server.

The email is able to flow securely and seamlessly through existing email infrastructure to recipients of the distribution list. Grace is on the distribution list so she receives the email from Frank.

(6) Grace receives the email. Grace’s client provides the attributes necessary to comply with the policy which includes her level 3 encryption certificate to the PDEP.

(7) Once Grace has shown she passes the policy requirements, the PDEP releases the message CEK to Grace using her level 3 encryption certificate.

(8) Grace uses her smart card to open the message. She sees the message is signed by Frank and marked with both the Program X and Company Foo IP policies.

If Grace replies to the email from Frank, the new message inherits the policy from the original message. If Grace includes some information which is Company Bar’s IP she also adds her company’s IP protection policy requirements to the message.

Frank receives the reply from Grace. Frank is able to prove his identity to the level requested by Grace and provides the requested attributes about himself to satisfy both the Program X export control, the Company Foo IP protection policies, as well as the Company Bar IP protection policies. Frank opens the email.

The policy also applies to messages forwarded by Frank and Grace because they contain information from Company Foo and Company Bar and both companies want consistent policy enforcement on their information.

After some time, Company Bar fails an audit to show they are complying which all the requirements for Program X. As a result, Company Foo updates its policies for Program X to remove Company Bar as an entity approved to access Program X data. Grace will no longer be able to access the Program X email as she can no longer satisfy the Program X policy requirements.

3.4.2 Regulated Email Requiring an Integrity Policy

Company Foo has been awarded a contract to build some equipment (Program X). This equipment is regulated by the National Aviation Authority (NAA) that has oversight of Company Foo. The NAA requires strict procedures at a number of significant events for Program X such as in the design and maintenance of the Program X (e.g., when a design is complete and released to manufacturing). The sign-off
Company Foo has instigated an email-based sign off procedure to simplify sign-off and reduce costs. It also has authored a policy for compliance with the NAA requirements. At the appropriate time, signoff email is sent to the designated program members. Recipients apply the NAA policy when they reply to the sign-off request message.

Frank is the lead on the Program X design team. They have a design which they believe can be released to the integration team. Frank initiates the sign-off process for the design.

Grace is one of the sign-off design team members for Program X. She receives the sign-off email. Grace responds and applies the sign-off signature policy to the email. The policy requires Grace to authenticate with the required level of assurance, present attributes about herself, her work effort assignments and professional qualifications to demonstrate compliance with the policy to send the message. The message is signed to indicate Grace met the policy.

When Frank initiates a Program X sign-off email the system must ensure the following objectives:

(a) Frank was authenticated to the level of identity assurance required under the policy to initiate the sign-off process.
(b) Frank possessed the necessary attributes as required by policy to initiate the sign-off process.
(c) The contents of the email are accurate to the level of integrity assurance required by the policy.
(d) Frank was fully aware and intended to initiate the sign-off process.
(e) The state of Frank's system was known to the level of assurance required under the policy to be free from agents which might interfere with the sign off process.
(f) Recipients can easily confirm over the lifetime of the design as required by the policy that the sign-off process met the policy without having to know the specifics of what the policy entailed.

The sequence of events Grace would use is as follows:

(1) Grace receives the sign-off request email.
(2) Grace replies to the email and completes the form data in the email to show she is approving the sign-off.
(3) Grace clicks the send button to send the email.
(4) Grace receives a sign-off confirmation dialogue before the email is sent where she is able to confirm her intent is to approve the sign-off of the component.

Grace’s system submits the decision request to send the sign-off email. Her system is asked to provide data about Grace, the state of her system and the data being authenticated. If Grace’s request meets the policy, her system receives a signed statement that the message meets the policy which is attached to the email and the message sent.

3.5 Delegation of Access to Email

There are a number of times when others are given access to a recipient’s mailbox or email is forwarded to other recipients based on the original recipient’s rules. This may be a long-standing relationship such as when an assistant is given access to an executive’s mailbox. Alternatively, it may be a temporary relationship due to short-term needs (e.g., to cover for a vacation). There are also organizational role mailboxes where the recipient is a role and one or more users are assigned to the role.

Grace is going on vacation. While Grace is away, Brian will act as a delegate for Grace. Grace configures a mailbox rule to forward Program X email to Brian for the duration of her vacation. Brian is able to satisfy the policy requirements for the Program X email as outlined above and is therefore able to open the protected email sent to Grace. Frank does not need to take any actions to allow Brian to access the email.

3.6 Regulated Industry Email

Some organizations work in areas which are intrinsically subject to policy such as regulatory policy, e.g., healthcare. In such environments the policies are often tied to the roles of the participants, the institution they are working at, and the subject of the exchange.

Hanna is a primary care physician working for FooBar Healthcare. She has a patient whom she is referring to a specialist at another healthcare facility for further diagnosis. Ida works as a specialist at the Bar Hospital. Hanna needs to send the relevant patient notes, test results, and comments to the specialist at Bar Hospital. Hanna knows she needs to comply with the confidentiality regulations and needs to respect her patient’s consent decree for the privacy of her healthcare information. When Hanna sends the referral message she must ensure:
(a) Only recipients who meet the healthcare regulatory policy and the patients consent decree can read the email.
(b) The message has the appropriate level of integrity protection and includes the data required by recipients to establish the data origin of the message as required by the policies.
(c) The recipients authenticate with an acceptable level of assurance (i.e. level 3 or above).
(d) Recipients present attributes about themselves necessary to verify compliance with the policies (e.g., their professional qualification, professional registration, affiliated healthcare facility and department).
(e) Recipients can verify the email is from the sender (Hanna) to the level of assurance as defined by the message policy (i.e., level 3 or above).
(f) Recipients can verify the email has not been tampered with to the level of assurance as defined by the message policy.
(g) Recipients are made aware that the message is a patient referral and contains sensitive patient data.

The sequence of events Hanna would use is as follows:

1. Hanna composes the email and adds an email address for the relevant department at Bar Hospital as a recipient of the message. She includes the patient information, test results, and comments in the email.
2. Hanna’s email client allows her to select a policy which is appropriate for her work.
3. Hanna selects the Patient Referral and Patient Consent Decree policies from the list of available policies.
4. The email client knows the protection to apply to the email, then encrypts the message.
5. Hanna clicks the "send email" button. The client then applies the protection required: it integrity-protects the message by signing with the private key on Hanna’s smart card, then encrypts the message and sends the CEK and the policies selected by Hanna to the PDEP server. It receives back protected policy metadata which contains the message CEK and the list of policies. It attaches the policy metadata to the message and sends the message to the email server.

The email is able to flow securely and seamlessly through existing email infrastructure to the message recipients. Ida is on the distribution list for the specialist department at Bar Hospital so receives the email from Hanna.

6. Ida sees the secure message from Hanna. Ida’s client provides the attributes necessary to comply with the policy, which includes her level 3 encryption certificate.
Once Ida has shown she passes the policy requirements, the message CEK is released to Ida using her level 3 encryption certificate.

Ida uses the private key on her smart card to open the message. She sees the message is marked with both the Patient Referral and Patient Consent Decree policies.

3.7 Email Compliance Verification

Verification is an essential part of compliance. Verification may be conducted by internal staff or external auditors. The verification need to confirm that the policy rules are being enforced. Auditing relies on the generation of artifacts to capture information about events. Typically, this is done via some form of logging. A challenge here is that for distributed system, the set of logs which completely describes the transaction are scattered across many systems so consistency of the audit settings and correlating all the audit data is problematic. Another consideration is accurately capturing only the set of desired data, i.e., accurately targeting the set of events that needs to be logged.

Jerry is the compliance officer for Company Foo. He has a procedure for ensuring compliance for Program X. The procedure defines what to log and when to audit access to Program X data. Jerry has tools to collect the audit data and run an analysis to verify the polices are being followed.

The sequence of events Jerry would use is as follows:

(1) Jerry configures an audit obligation for access to Program X data. The obligation defines the set of attributes to capture when Program X data is accessed. The obligation is part of the Program X policy. Part of the Program X policy is the set of PDEPs which can process policy decisions on Program X data.

(2) Jerry configures his audit log collection to download Program X audit log entries from the designated PDEPs.

(3) Jerry also has an audit confirmation tool which "pings" the PDEPs for access to Program X data. Jerry’s audit log analysis tool looks for these pings to confirm that auditing is taking place as expected.

3.8 Email Pipeline Inspection

Organizations have a huge incentive to inspect emails entering or leaving the organization. Such inspection is desired for many different reasons. Inspection of mail leaving an organization is targeted towards making sure that it does not leak confidential
information. It also behooves organizations to check that they are not a source of malicious content or spam. Inbound mail is checked primarily for malicious content and phishing attempts as well as spam. For domains with a high volume of messages there is a strong need to process email with minimal overhead. Such domains may mandate that they be pre-authorized to process an email due to the overhead a per-message request to an external service would add to message processing.

Company Foo has a policy to scan all inbound and outbound email to ensure it is free from malware. Company Foo also wants to ensure email is not spam. Company Foo can own their scanning servers or such checks may be outsourced to a third party service. Company Foo wants to ensure that its policy of scanning message contents also applies to encrypted email.

The ability to decrypt and check the message content for malicious content is highly desirable. There are a number of methods that can accomplish this:

1. When a Company Foo client requests to send a Plasma email, the PDEP is able to check to see if the policy allows email content inspection by MTA for this policy, and if it does, that Company Foo has an outbound email scanning, and that the scanning servers meet the policy requirements. It is able to pre-authorize the Company Foo email scanning servers to access the email.

2. The scanning MTA authenticates to the PDEP as an entity doing virus and malware scanning on a protected message. If the PDEP has specific policy that allows for access to such a scanning MTA service, the appropriate decryption keys will be released and the server will scan the mail and take appropriate action.

3. The policy server is configured with information about various gateways (both internal and external) and has certificates for the known gateways. The policy server can then return a normal X.509 recipient info structure (cryptographic lockbox) to the sender of the message for direct inclusion in the recipient info list of the message. This allows normal S/MIME processing by the scanning MTA without the necessity to query the PDEP server for keys for specific messages.

4. If the scanning MTA server cannot gain access to the decrypted content using one of the two proceeding methods, it either passes the encrypted mail on to the recipient(s) without scanning it or it rejects the mail. This decision is based on local policy of the scanning MTA. If the message is passed to
the recipient(s), then the necessary scanning either will not be done, done by a downstream MTA, or done on the recipient’s system after the message has been decrypted.

3.9 Distribution List Expansion

A distribution list (DL) is a function of an MTA that allows a user to send an email to a group of recipients without having to address all the recipients individually. The membership of the DL may be confidential so the sender may not know all the recipients. The DL may be maintained by an external organization. Since a DL is identified by an email address, the user may be unaware they are sending to a DL.

Plasma polices may have the list of recipients as a parameter, thus the fact that the message is being process by the distribution list means the MTA processing the message needs to update the policy to allow the new recipients to access the message. Organizations may also require inbound scanning of email and have thus published keys to enable pre-authentication of the MTA by the sender to expedite processing. For both scenarios the DL MTA has to notify the Plasma server that it is adding recipients to the message and supply the list of new recipients. The Plasma server can then take appropriate action on the message token and return an updated token if required.

3.10 Scalable Decision Making

Collaboration involves working with external organizations; e.g., partners and suppliers. These collaborations may be short or long-lived, with a small or very large number of participants. Organizations therefore need flexibility in deployment and scaling. Organizations do not want to be forced into having to provide capacity themselves for all decision making over their data. Senders would be happy to delegate decisions where appropriate to partners or external services provided those decisions use the rules they define for their data. Likewise, recipients might be happy to leverage their local decision capacity providing they don’t have to duplicate the rules of the partners, and can simply and easily use policies published by their partners. An organization may also want to use cloud based PDEPs where appropriate as a cost effective way to add capacity and to be able to respond to transient capacity fluctuations.

See section 3.4.1 for a description of the scenario.

The Program Managers for Program X at Companies Foo and Bar agree to a series of roles which are used to manage personnel and their
assigned policy groups. The policy administrators for Company Foo and Bar respectively publish the roles and a policy collection for each role. There are rules associated with the policy collection, for example every role uses the Program X policies published by Company Foo. Employees from Company Foo also get the Company Foo Intellectual Property policies for those roles, whereas employees from Company Bar get the Company Bar intellectual property policies for Program X. Company Foo has also decided to allow enforcement of Program X policies by decision engines in both Company Foo and Company Bar. Company Foo has also decided to use a cloud-based decision engine for Program X to allow lower cost capacity and scaling. Company Foo is able to add new instances of the cloud-based decision services as the program scales up and more uses start working on the program. Each decision engine dynamically discovers the policies it needs from the set published by Company Foo and Company Bar. Both Company Foo and Company Bar can add new policies to the policy collections at any time and they are dynamically discovered by all the policy decision engines.

3.11 Related Use Case Scenarios

There are other use case scenarios which are related to the email cases because they would be subject to the same policy requirements. Email allows users to create content and transport it to a set of recipients. Similar use cases to the above can be performed with other data formats such as documents and instant messages. Policy is tied to the information so is agnostic to the underlying data format therefore if an organization has a policy relating to a type of information, then that same policy would apply to the same information in an email, a document, or an instant message, etc.

3.11.1. Document Protection

This use case scenario is very similar to 3.4 and 3.6 above. The difference is that the information being generated is in the form of a document not an email. It could be as part of an ad-hoc sharing or a regulated sharing of information.

Frank is an employee of Company Foo. He has been assigned to Program X. Grace is an employee of Company Bar. She has been assigned to Program X. Frank creates a document for the program. He also includes some Company Foo IP in the document. When Frank creates the document he must ensure compliance with export control regulations and his corporate IP protection policies. Frank must ensure:

1. Only users who meet the Program X policy or Company Foo’s intellectual property protection policy can open the document.
2. Users authenticate with an acceptable level of assurance as defined by the set of policies applied to the document.

3. Users present any other attributes about themselves necessary to verify compliance with the applicable policies.

4. Users can verify who the author was to an acceptable level of assurance as defined by the document policy.

5. Users can verify the document has not been tampered with to an acceptable level of assurance as defined by the document policy.

6. They can also tell it is a Program X document and that the contents can only be shared with other Program X workers.

Frank creates a document for Program X. He includes some information related to Program X. Frank also includes some information which is Company Foo’s IP.

Franks word processor application allows him to classify the document. Frank classifies the document as Program X and Company Foo proprietary information.

The word processor application knows the protection to apply to the document; it integrity-protects and encrypts the document; it sends the CEK and the policies Frank selected to the PDEP; the PDEP returns protected Policy metadata containing the CEK and the list of policies. The word processor then attaches the policy metadata to the document.

The document is able to be published on a cloud-based Web portal. The document is protected while in transit to the portal and at rest on the portal. The document is also protected on any backup or replica of the portal data. Frank does not need to worry about where on the portal he publishes the document. He can make the most appropriate choice based on the project and the document content.

Grace sees the document on the portal and tries to open the document. Grace is able to prove her identity to the level required by the policies applied to the document and provides the requested attributes about herself to satisfy both the Program X export control and the Company Foo IP protection policies. Grace opens the document.

If Grace edits the document and includes some information which is Company Bar’s IP she adds her company’s IP protection policy requirements to the document. Grace saves the updated document to the same location on the portal.
Frank sees that Grace has updated the document on the portal. Frank is able to prove his identity to the level required by both the Company Foo and Company Bar policies and provides the requested attributes about himself to satisfy both the Program X export control, the Company Foo IP protection policies as well as the Company Bar IP protection policies. Frank opens the document.

### 3.11.2 Instant Message Protection

This scenario is very similar to 3.4 and 3.6 above. The difference is that the information being generated is in the form of an instant message not an email. It could be as part of an ad-hoc sharing or a regulated sharing of information.

Frank is an employee of Company Foo. He has been assigned to Program X. Grace and Hank are employees of Company Bar and also have been assigned to Program X. Frank wants to discuss an urgent topic with Grace and Hank. The topic necessitates discussion of Company Foo IP. Because of the urgency, Frank wants to use IM. Frank must ensure:

(a) Only users who meet the Program X policy or Company Foo’s intellectual property protection policy can join the IM session.

(b) Users authenticate with an acceptable level of assurance as defined by the set of policies applied to the IM session.

(c) Users present any attributes about themselves necessary to verify compliance with the applicable policies.

(d) Users can verify who the IM initiator is to an acceptable level of assurance as defined by the session policy.

(e) Users can verify the IM data has not been tampered with to an acceptable level of assurance as defined by the session policy.

(f) They can also tell the session is a Program X session and the contents can only be shared with other Program X workers.

The sequence of events Frank would use is as follows:

1. Frank initiates the IM session and includes Grace as a participant.
2. Frank’s IM client allows him to select a role which is appropriate for the session. Frank then selects the Program X and Company Foo IP policies for the session.

The IM application knows the protection to apply to the document; it integrity-protects and encrypts the message; it sends the CEK and the
policies Frank selected to the PDEP; the PDEP returns protected Policy metadata containing the CEK and the list of policies. The IM application then attaches the policy metadata to the message.

The IM is able to flow securely and seamlessly through the existing IM infrastructure to session participants. Grace is a session participant so her IM application attempts to join the IM session with Frank. Hank is in a meeting so does not join the IM session at that time.

(3) Grace receives the IM and sees it is a secure IM from Frank. Grace’s IM application provides the attributes necessary to comply with the policy which includes her level 3 encryption certificate.

(4) Once Grace has shown she passes the policy requirements, the she receives the IM session CEK protected by her level 3 encryption certificate.

(5) Grace uses the private key on her smart card to decrypt the session CEK and opens the IM session. She sees the from Frank is marked with both the Program X and Company Foo IP policies.

(6) Grace composes a response to Frank’s question and hits send.

(7) When Hank’s other meeting is finished, he sees the IM invitation from Frank and attempts to joins the IM session and is able to do so because he to meets the policy requirements and sees the messages from Frank and Grace.

4. Plasma Data Centric Security Model

A common theme from these scenarios is the need to closely tie the information asset to the set of technical controls via the data owner’s policies in such a way so it is possible to consistently apply the technical controls across a broad set of applications (not just email), for a broad set of users (not just those within an organization), and in a broad set of environments. Assumptions based on closed-world, enterprise security models are increasingly breaking down. Perimeter security continues to diminish in relevance and focus needs to be shifted to self-protecting data as opposed to protecting the machines that stores such data. The binding between the data and the applicable polices needs to happen as close to the data creation time as possible so ad-hoc trust decisions are not required.

The delivery of the documented use cases will require the integration of many existing and some new protocols. In order to ensure the right overall direction for Plasma as each part of the work proceeds, a high level data model is documented here to act as a guide. While this is technically informative to the developments of each individual component, it is normative to the work overall.
This Data Centric Security model is based on a well-established set of actors for policy enforcement used elsewhere [RFC3198] [XACML-core].

Figure 1 shows the relationship between the actors.

The Plasma model is applicable to any type data (email, documents,
databases, IM, VoIP, etc.). This is to facilitate consistent policy enforcement for data across multiple applications. Another objective is to not require the data holder to have access to the plain text data in order to be able to make decision requests to the PDEP. The policy decision is complex so the content creation DR in Plasma just uses policy pointers or labels to indicate the set of policies applicable to content. The content consuming DR dynamically discovers the PDEP's that are authoritative for the decisions on protected content in question. The PDEP’s dynamically discover the specifics of a policy from a PAP using the policy references. The specifics of policy authoring and policy decision logic modules are matters beyond the scope of this document. It is important to note that the actors in this model are logical entities and as such can be combined physically in different configurations.

- The Plasma model uses references to bind the data and the policy. When information is created, it is encrypted and a list of policies that must be enforce by the PDEP is bound to the protected data.

- The Plasma model includes policy discovery capability for subjects. This enables subjects to interact with one or more PDEP to discover the set of polices the set of polices each PDEP would permit the subject to use to protect new content. The PDEP issues a role token to subject which contains one or more policy collections. Each policy collection is identified by a role name. Subjects can pick any combination of polices from a policy collection, but cannot mix polices from different policy collections. The token issued to subjects containing the policy collections is known as a role token.

- The Plasma model is an Attribute-Based Access Control (ABAC) model where the ABAC policy is specified in terms of a set of attributes, their values, and their relationships. The policy may specify attributes about the subject, their device, or their environment, or attributes about a resource.

- The ABAC policy does not require the subject provide their orthonym. Subjects could be anonymous or pseudonymous. What is required is the presentation of a set of attributes that satisfies the policy.

- The subject can be required to bind the supplied attributes to the channel with the PDEP to a level of assurance as required by the PDEP. If the PDEP only requires low assurance, bearer tokens over TLS would be suitable. If the PDEP requires higher assurance, then the holder of key tokens over TLS would be required where the token key is bound to the TLS channel.
This model also supports Capability-Based Access Control (CBAC) where security tokens represent a capability to meet a policy. Once a subject has proven compliance with a policy, they can be issued a capability token. The client can subsequently present this capability token in lieu of a token or tokens with the set of subject attributes. The net result is that the model can transition to a Capability-Based Access Control because the capability token is an un-forgeable token of compliance with a policy. The token can be used with any resource tagged with the same policy.

Plasma has a baseline of a secure transport between the DR and the PDEP. One of the decisions the PDEP has to make is the level of assurance on the release of the CEK to the subject. For example, the PDEP can release a clear text CEK over the secure transport to the DR. Alternatively, the PDEP could require the production of a high-assurance X.509 encryption certificate as a subject attribute to generate an encrypted CEK.

For the purpose of the Plasma work, it is desirable that the DR and PDEP be clearly defined as separate services which may be on separate systems. This allows for a generalization of the model and makes it less dependent on any specific deployment model, policy representation or implementation method. It also allows for a greater degree of control of the PDEP by an organization such that it is possible to keep all of the PDEP resources directly under it’s control and independent of the data storage location.

The base set of information for a Plasma client is as follows:

- The address of one or more IdP(s) able to issue identity attributes to the subject
- A means to authenticate to the IdP(s) and issue attributes to the subject
- The address of zero or more AtP(s) able to issue additional attributes to the subject
- The address of one or more Plasma PDEPs able to issue role tokens to the subject to initiate Plasma policy discovery.

From this base set of data, the subject is able to authenticate to each Plasma PDEP in turn using the identity token from the IdP and discover the set of assigned roles. Each role has a set of policies which can be applied to data. A subject may be assigned to multiple...
roles and therefore has the ability to select the most appropriate role for the content being created. Once a role is selected, the subject is able to choose one or more policies from the policy collection for that role. Role assignment is dynamic so the role discovery needs to be done on a regular (but not frequent) basis. Policy selection during content creation can be either manual or automatic. A DR may have sufficient context to be able to select the role and policies for the subject or have some rules that facilitate policy selection.

The model allows the content creation DR to discover the role assignments from multiple PDEP which would allow the subject to access policies based on roles from within their organization and from any partner organization due to cross-organization collaboration. The PDEP’s that are authoritative for the role assignment for a subject may be different from the PDEP that are authoritative for enforcement of a policy collection in question. The DR uses the role token to authenticate the content creation request. The PDEP will check that the requested list of policies for the information is a subset of the policies in the role token. If the set of policies is a subset of the policies in the role token, then it will issue the policy metadata token to be attached to the protected data.

The policy metadata token is an signed data structure created by the PDEP which is bound to the protected data. It contains public policy metadata attributes which are used by the DR. An example of a public policy metadata attribute is a list of one or more URLs which represent the PDEPs that can make policy decisions using the policy metadata token. The DR can submit the decision request to any PDEP in the list. The policy metadata token also has a confidential payload containing private policy metadata attributes used by the PDEP to make policy decisions. An examples of a confidential policy metadata attribute is the list of CEKs for the protected data which would be released to the DR if it passes the policy checks.

Policy rule processing and distribution is complex, so the Plasma model does not require policy rules to be distributed to the DR. The DR submits the policy metadata token as part of the decision request. The confidential portion of the policy metadata token contains a logic tree of policy references. The PDEP uses the policy references to discover the policy rules to apply to the request. The logic tree defines the relationship between the polices. The tree has a series of nodes where each node represents a set of polices and the relationship for the polices at the node e.g., are they combined via AND clause or an OR clause. The pinnacle of the tree represents the decision from all the polices in the tree. The use of policy references minimizes any policy maintenance issues relating to the
protected data due to policy updates. The policy rules can be updated
and the new rules discovered on subsequent decision requests.

The DR and PDEP are required to carry out obligations of the policy
such as specific encryption requirements, e.g., key size or
algorithm, data integrity requirements, time-to-live of the CEK, or
audit record creation requirements. It is a matter for the policy on
how to determine if the DR or PDEP is trusted to carry out the
obligations. This could be achieved by devise type and state
attributes.

The PDEP makes its decisions based on the requested action from the
DR, the policy requirements from the PAP(s), and the information from
the PIP(s) about the subject, the subject’s device, and the subject’s
environment. The information about the subject may be exchanged
directly between the PIP(s) and the PDEP (Back End Attribute
Exchange) or indirectly via the DR (Front End Attribute Exchange) or
both. The content creator can also include attributes in the policy
metadata.

There is no guarantee that identity and attribute providers will
consistently use the same name to identity a specific attribute or
attribute data. For example they may use different schemas to
identify an email address or use localized names to describe job
functions or roles. These kinds of values may be standardized within
communities of interest, but not globally across all identity and
attribute providers. Therefore it is necessary to canonicalize the
attribute names and values before processing by the policy. The
attribute name and value mapping is part of the policy data set,
_i.e._, it is in addition to the policy processing rules.
When drawing a line where the actors in the model are full trusted with the clear text data there are three possibilities (see figure 2).

Figure 2a shows the full trust line between the user application and the Policy Enforcement Point (PEP). This is the model for current standard access control mechanism, e.g., XACML [XACML-core]. In 2a, the PEP has full access to the plain text data. It makes decision requests to the PDP and if the decision is affirmative, allows the PEP to release the data to the application. To use figure 2a for secure email would require every MTA and MUA to be fully trusted with plain text data which is impossible.
Figure 2b shows the full trust line between the PDEP and the DR. In 2b, the DR only has cipher text data. The data is encrypted with a content encryption key (CEK) and the PDEP has access to the CEK. The PDEP releases the CEK to the end-user application when access is granted so the application can recover the plain text. This mode is viable for secure email as it does not require the MTA to be trusted with the plain text data and either the MTA or MUA can act as a DR.

In figure 2c, no actor is given full trust. When the data is encrypted, the CEK is encrypted for each recipient just as S/MIME does today. The encrypted CEKs are given to the PDEP and the PDEP releases the encrypted CEK when access is granted. This mode is also viable for secure email as the sender can use either conventional public key cryptography or Identity-Based Encryption[RFC5408] to protect the CEK for each recipient.

4.1 Plasma Client/Server Key Exchange Level of Assurance

There are a number of mechanisms by which a client and servers can exchange CEKs. As a baseline, Plasma is establishing a secure transport between the client and server via TLS. However the client may be a proxy acting on behalf of the subject, therefore transporting a clear text CEK over the TLS transport would expose the key to the proxy. There also may be a proxy at the server which is terminating the TLS transports and forwarding the requests to another server which would mean a clear text CEK sent over the transport would be exposed to the server proxy. Policies may require a higher level of assurance that the CEK is not exposed to unauthorized principals. This requires encrypting the CEK for the subject before transport. This would further require the client or the server to provide a public key to the other party to be used to protect the CEK before sending it over the secure transport.

4.2 Policy Data Binding

There are three ways to bind policy to data:-

- By value. This is where a copy of the machine-readable rule set is directly associated with the data, e.g., where a file system has an Access Control List for the file or directory, or where a rights management agent embeds a copy of the policy expressed in a policy expression language in the rights-protected data. When an access request is made to the data, the PDEP compares the access request to the policy on the data itself.

- By reference. This is where a reference to the policy is directly associated with the data, e.g., a URI or a URN which identifies the policy to be enforced or points to where the policy is published.
For example with S/MIME, the ESS label identifies the applicable policy by an OID. When an decision request is made to the data, the PDP finds the policy based on the identifier and then compares the access request to the referenced policy.

- By inference. This is where the policy has a target description in terms of resource attributes the policy applies to. When a decision request is made, a set of attributes describing the resource which is the subject of the decision request is included in the request by a PEP. The PDP then compares the resource attributes to the set of target descriptions of the policies in its policy store to determine the set of policies to apply to the request. For example when an XACML policy is authored, a target description in terms of the attributes of the resource for the policy is also defined. When an XACML decision request is made, the PDP finds the policy set to apply to the request by matching the set of attributes in the request against the target description associated with the policies in its store. It then processes the decision request using the identified policy set.

The chief strength of binding policy by value is its simplicity. The policy, being local to the data, can easily and quickly be read by the PDP. The chief weakness in binding policy by value is maintaining policy over time as binding by value results in the policy being replicated for every instance of data the policy is applied to. Many policies have a multi-year life span and over the course of time, there is a very high probability that the policy would need to be updated. Given the high number of copies, updating a value-bound policy has proven to be a very costly and imperfect process both from an enforcement and audit perspective. This process is complicated by the fact that because only the result is stored and not an identifier, it is hard to identify the policy that has to be updated.

The chief strength of binding by names is that once the policies are bound to the data, the same policies continue to be applied regardless of PDEP configuration or state. These policies may change their rules over time, but there is no doubt which policies would be enforced on the data. Another strength of binding policy by reference is it has a clear result as to the set of policies the PDEP has to apply. It the PDP does not have a policy, the reference allows the PDEP to discover the missing policy. If the PDEP is unable to access a policy for whatever reason, it knows to fail the decision request with a different error, i.e., "don’t know", which means the DR can reasonably try other PDEPs. The chief weakness in binding by name is adding or removing policies requires updating the policy metadata. Adding or removing policies has the same difficulties as maintaining policies by value.
The chief strength of binding by inference is it can often be applied to data without impacting the storage format providing the data already has rich and well defined set of metadata such as the structural metadata of an SQL table. It also allows new policies to be applied to the data without updating the metadata. Unstructured data such as documents have the ability to store metadata but the challenge here is what metadata to capture. The nature of the metadata is also context specific, e.g., the policy target description required to match structural metadata from an SQL query would be different from the policy target description for matching content metadata for a document. The chief weakness in binding by inference is the reliability of the matching of the metadata to the policy target description. There are a number of factors which affects the policy matching process:

* The set of available metadata varies with different data types which makes the policy target definition more complex, e.g., structured data such as SQL databases have structural metadata whereas unstructured data such as documents have content metadata.
* There is a relationship between the metadata you need to capture and the policies you need to enforce. It’s therefore hard to generalize the rules for what metadata is necessary independent of knowing what metadata policies require.
* The resultant set of policies to enforce for a decision request is dependent on the PDP having a complete the set of policies. It is impossible, however, to detect missing policies based on the request. Likewise, it is also impossible to detect if erroneous policies have been selected based on the request. If data moves from store to store and thereby uses a different PDPs, it’s impossible to determine the correctness of the result of the policy matching process by the new PDP.

The Plasma model is choosing to use binding by name for two reasons:

1. The overarching need to consistently enforce the policies selected at creation time over the lifetime of the data. The typical use case is that the set of policies to be enforced on the data may change their rules over time but it is the same set of policies that are enforced over the lifetime of the data.
2. Data in many cases is mobile and travels between users and organizations. Any dependency on consistency of the decision making entity would be difficult to enforce or verify.

4.3 Content Creation Workflow
The content creation DR bootstraps itself via the following sequence of events:

(1) The content creation DR is configured with the set PIPs and PDEPs it trusts.
(2) The content creation DR submits a request for a role token to all the trusted PDEPs. The role token defines the set of roles the PDEP allows for the subject. The subject is authenticated and the contents of the role token authorized by the PDEP via attributes from the PIP(s). The PIP attributes can be obtained by the PDEP either via front-end (relayed to the PDEP from the PIP via the subject) or back-end (direct exchange between the PDEP and the PIP) processing.
(3) The content creation DR receives zero or more roles tokens from each of the PDEP. Each role token has a one or more policy collections defining the set of allowed policies for that role when creating new content.

The DR is now initialized with a list of roles and role tokens. It is now ready to create content and request protection of that content from PDEPs. This role token request process would typically be performed as part of the application initialization process. Role tokens can be cached to reduce the number of times the application has to invoke the role token request process. When the user wants to create new content, they use the following sequence of events:

(i) The user creates the new content
(ii) The user selects the appropriate role for the content, then selects one or more policies from the policy collection that are applicable to the content. When the content creation process is complete, the DR:
(iii) Encrypts the content with one or more locally-generated CEKs
(iv) Submits a policy metadata token request to the PDEP together with the CEK(s), the set of required policies to be applied, the role token from the PDEP, and the hash of the encrypted content. The CEK(s) in the request can be either raw key(s) or CEK(s) encrypted by a KEK if the policy does not allow the PDEP to have the ability to access the plain text data.
(v) The PDEP verifies the set of requested policies is a subset of the policy set in the role token. In addition to the role token, the PDEP may also require any other attributes from the subject as defined by policy to process the creation request.

If the request satisfies the policy requirements, the PDEP generates the encrypted policy metadata which contains the list of policies and the CEKs. The metadata is encrypted by the PDEP for all the PDEPs allowed to make decision requests.
for the data (the content creation PDEP does not have to be in the set of PDEPs allowed to make access control decisions). The PDEP includes a list of URLs for all of the PDEPs allowed to process decision requests and the hash of the protected content as signed authenticated attributes in the policy metadata token, then it signs the encrypted metadata.

(vi) The PDEP returns the policy metadata token to the DR
(vii) The DR attaches the policy metadata token to the protected content and distributes the content.

4.4 Content Consumption Workflow

When a user wants to open some protected content they would use the following workflow.

(a) The DR verifies the certificate in the signed policy metadata then determines via local policy if it wants to process the protected information based on the identity of the PDEP.
(b) The DR verifies the signature on the policy metadata token and the binding to the encrypted data by hashing the encrypted information and comparing it to the authenticated attribute in the policy metadata
(c) The DR creates read token request. The request contains the signed metadata from the content together with one or more authentication tokens issued by a PIP. The request may also contain attributes about the request such as the purpose of use of the data.
(d) The DR sends the read token request to one of the URLs of the PDEPs in the authenticated attributes of the signed metadata
(e) The PDEP decrypts the policy metadata, de-references the policy pointers, and determines the set of rules to apply to the request based on the policy published by the PAP. The PDEP then determines the set of attributes it needs to evaluate the policy rules. The PDEP can use PIPs it has direct relationships with to query attributes about the subject. If the PDEP is missing attributes it need to process the policy, it returns a list of the missing attributes to the DR.
(f) If the DR receives a list of missing attributes from the PDEP, it obtains the missing attributes requested by the PDEP from a PIP and sends them to the PDEP in a new read token request.
(g) Once the PDEP has a complete set of attributes, and the attribute values match those required under the access policy, the PDEP releases the CEK to the DR along with a TTL which defines how long the DR can use the CEK before it must discard the CEK and reapply for access.
(h) Once the DR has the CEK it decrypts the information. It caches the CEK until the TTL expires.
4.5 Plasma Proxy Servers

There are two separate use cases for proxy servers in Plasma. The forward proxy use case where a DR client needs to connect to a PDEP outside of its organization and the reverse proxy use case where a DR client outside an organization, needs to connect to a PDEP.

A recipient has no control over senders creating Plasma email (or any other type of Plasma protected content) and sending it to them. Malicious senders can craft harmful payloads and protect it in a Plasma envelope. Therefore, Plasma recipients need a policy to determine the set of Plasma DPEP services they are willing to interact with. This can be a local policy i.e., a policy for the allowed set of PDEPs a DR client can interact with. This policy would need to be distributed to every DR client. An alternate approach is to have a forward proxy manage the policy on behalf of the DR client. A forward proxy would eliminate the need to distribute policy by mediating the connection requests from the DR clients to the PDEP services. The forward proxy could be a server belonging to the DR client organization or a cloud service.

In the no-proxy use case the DR client would connect via TLS directly to the URL contained in the policy metadata. The DR would thus need local policy to determine whether to connect to the PDEP URL. If a forward proxy is preset, the DR client would attempt to connect via TLS to the forward proxy. The forward proxy would then connect to the PDEP if its policy allowed.
Since the Plasma service has sensitive cryptographic keys used to protect the data CEKs, it would be unwise to host those servers directly connected to the Internet. However, PDEPs will need to be Internet addressable for requests from DR clients outside the organization. The simplest possible configuration would be to have a passive reverse proxy in front of the Plasma server. Since Plasma is using TLS, a passive proxy cannot inspect the data inside the TLS session. The passive proxy has therefore a limited function and would be only able to filter based on session characteristics e.g., source IP addresses. The Plasma protocol is a series of request-response messages, so an active reverse proxy can be implemented like other store-and-forward message based services (e.g., SMTP). The Internet-facing proxy server would terminate the TLS connections from the external DRs. The active proxy can then scan submitted requests to ensure they are not malformed and are free from malicious content before relaying messages to a full PDEP server further inside the network for processing of the request.
4.6 Policy Types

Policies range from very simple to very complex. Policies have dependencies not only on the technical implementation of the software but on the range of attributes a PIP would issue to subjects. This is likely constrained by the physical procedures a PIP could support to capture and verify the information about the subject. To manage this range of requirements, this model uses two type types of policy.

4.6.1 Basic Policies

Basic policies are intended to be universally usable by employing a small, fixed set of attributes that are available from all PIPs. For example, basic policies are intended to be equivalent to sending encrypted email with S/MIME today i.e., authenticated recipients of
the email get access to the message. Basic policies target scenarios involving consumers and small businesses who are using public PIPs which issue a limited set of attributes. It is expected that all Plasma clients and commercial IdPs would be capable of supporting basic policies due to the finite set of attribute set required which will simplifies development, testing and deployment. Later standards may expand the set of attributes supported by basic policies and hence define richer basic policies.

4.6.2 Advanced Policies

Advanced policies are intended to be used where one or more policies are required on the content that require an expanded set of attributes from an IdP. They are intended to target more complex policy requirements such as content with regulated information or content subject to organizational and contractual policies. The input set of attributes are defined by the policies. These attributes are, in theory, unbounded and can be either primordial such as date of birth, or derived attributes such as age, or both. In practice, advanced policies are constrained by the set of attributes available under the IdP Trust Framework for the subjects. A data object may require multiple policies and any instance of multiple policies requires a logical relationships between the policies, e.g., they can be AND-ed or OR-ed together. It is not expected that all Plasma clients will support the rich set of attributes necessary for advanced policies.

5. Message Protection Requirements

5.1. General Requirements

Confidentiality policy protected data MUST be protected from unauthorized disclosure, protected from unauthorized alteration and provide data origin authentication.

Integrity policy protected data MUST be integrity protected from unauthorized alteration and provide data origin authentication.

Every authentication has a level of identity assurance associated with it depending on attributes such as the identity checks made about the subject and the authentication technology used by the subject. The authentication of content creators and content consumers MUST support the multiple levels of identity assurance framework (see sections 3.1, 3.2, 3.3, and 3.4.)

The specifics of every possible authentication mechanism or every detail about how the subject’s identity was proofed by the IdP cannot be known to the DR and PDEP, therefore the specifics of how the
sender or recipient achieves the required level of identity assurance
MUST be abstracted from the PDEP and DR by use of a simple numeric
scale, e.g., 0-n where n is linked to an identity assurance framework
that defines the specifics of how to derive the LoA (See sections 3.1,
3.2, 3.3, and 3.4.)

Access policies are complex and subject to change over time. For
this reason, policies MUST be identified by reference rather than
inclusion of the actual policy with the data so the policy change can
be implemented without updating the data. (See section 3.4.1.)

Access to the plaintext of the content MUST only be provided after
the recipient has either provided suitable valid attributes to the
PDEP or the PDEP finds attributes about recipient directly from a
PIP, thus satisfying the policy as defined by the sender. (See
sections 3.1 3.2, 3.3, 3.4.1, and 3.5.)

The sender MUST be provided with a list of policies applicable to
content they create and scoped to their current role, i.e., what
tasks they are currently assigned to deliver. (see sections 3.1, 3.2,
and 3.3.)

The specifics of the access control policy used by the PDEP MUST be
abstracted from both the sender’s and the recipient’s DR, i.e., the
DR MUST NOT make the access control decision or need specifics of the
access policy requirements. (See sections 3.1, 3.2, 3.3, and 3.4.)

A content consumer DR MUST receive authenticated attributes of the
identity of the creator, the level of identity assurance of the
creator, and the cryptographic fingerprint of the original content so
that the DR can confirm who created the content and that the content
has not been altered. (see sections 3.1, 3.2, 3.3, and 3.4)

The key exchange between content creator and content consumer and the
PDEP MUST support multiple levels of assurance so an appropriate
strength of mechanism can be selected based on the level of assurance
required. For example, for low assurance situations this could be via
a plan text CEK over a secure transport such as TLS. For high
assurance situations, the recipient MAY be required to provide a
suitable key exchange key such as an X.509 certificate to encrypt the
CEK. (see sections 3.3 and 3.4)

The level of key exchange assurance required MUST be selected by the
sender’s policy and enforced by the PDEP. (See sections 3.1, 3.2,
3.3, and 3.4.)

If the recipient is unable to initially comply with the sender’s
policy, then if they are subsequently able to get the required
credentials or attributes it MUST be possible for to retry access to the content without intervention from the content creator.

A time-to-live (TTL) MUST be provided to content consumers when access is granted by the PDEP to define when the DR MUST discard the message CEK and submit a new access request to the PDEP. The TTL value MUST be based on the message policy and optional attributes about the content consumer and its environment.

The PDEP MUST be stateless for processing policy requests from content creators and consumers with respect to any instance of protected content. It MUST be possible to have multiple instances of a PDEP service and load balance requests across all instances of the service transparently to the client and not require synchronization of state about requests between instances of the service.

A PDEP MUST be capable of generating audit events associated with access to protected content using policy defined by the PAP.

5.1.1 Email Specific General Requirements

It MUST be possible for domains to publish keys and attributes about the boundary inspection agents. This allows senders to pre-authorize the inspection agents of recipients for access to messages.

It MUST be possible for MTAs to request access to protected messages for which they have not been authorized by the sender (See section 3.8).

Is should be possible for an MTA to pre-authorize another to access a protected message (See section 3.8).

5.2. Basic Policy Requirements

The use of a Basic policy MUST be backwards compatible with existing S/MIME.

A sender’s agent MAY discover some recipients’ encryption certificates and create recipient info structures using the existing S/MIME standard (unless specifically forbidden by the selected policy).

A sender’s agent MAY elect to use a Basic Policy mechanism for recipients for whom encryption certificates cannot be discovered.

Four Basic policies are to be defined by this work. These Basic policies MUST map to the LoA of NIST 800-63-1. This does not preclude other Basic policies to be defined by other groups, trust
When using a Basic policy defined by this work, the sending agent MUST define which Basic policy is required and the list of reciprocals.[RFC5322] recipients.

A sender using Basic policy MUST be able to send protected messages without discovering a recipient’s encryption key.

A sender using Basic policy MUST NOT require a bilateral agreement between sender and recipients as a prerequisite to sending the message.

5.2.1 Email Specific Basic Policy Requirements

The use of Basic Policy MUST be backwards compatible with existing S/MIME encryption.

A sender’s agent MAY discover some recipient’s certificates and create recipient info structures as per the existing S/MIME standard and elect to use the new mechanism for recipients it cannot discover keys for rather than remove the recipient’s without certificates.

5.3. Advanced Policy Requirements

A Basic policy MAY be combined with Advanced policies.

It MUST be possible to apply one or more Advanced policies to content.

Where two or more policies are applied to content, the logical relationship between the policies MUST also be expressed e.g., are the policies a logical AND or a logical OR. (See section 3.3)

An advanced policy MAY require attributes about:

- The content consumer
- The device the content consumer is using
- The environment of the device that is attempting to access the protected content
- The content being accessed

Advanced policy MUST support an extensible list of obligations on the DR or PDEP such as use of the policy requires some specific action on the part of the content creator, e.g., signing content with two-factor smart card and/or that the signature complies with the legal requirements for the transaction, or the signature needs to be able to be verified for an extended period. (See sections 3.3 and 3.4.)
Advanced policies MUST support the ability to verify the content for an extended period as required by policy. For example policy may require signatures to be verifiable for a period of 10 years.

Advanced policies MUST support the ability to resign the data to support the verification over the extended period.
6. IANA Considerations

This document describes the requirements for message access control. As such, no action by IANA is necessary for this document.
7. Security Considerations

Authentication by itself is not a good trust indicator for users. Authentication raises the level of assurance the identity is correct but does not address whether the identity is trustworthy or noteworthy to the recipient. Authentication should be coupled with some form of reputation e.g. the domain is on a white list or is not or a black list. Malicious actors may attempt to "legitimize" a message if an indication of authentication is not coupled with some form of reputation.

Malicious actors could attempt to use encrypted email as a way to bypass existing message pipeline controls or to mine information from a domain. Domain should have sufficient granularity of policy to handle situations where their email pipeline agents have not been authorized to inspect the contents.

It must be possible for a third party to, upon correctly presenting a legitimate legal justification, to recover the content of a message. This includes the Sender’s and Recipient’s companies for business continuity purposes, as well as Law Enforcement. If the entity requesting the information and the entity controlling the access are in different jurisdictions, then the process would be subject to some form of rendition.

The use of a security label type that requires the recipient of a message to query a PDEP in order to obtain the contents of a message opens an additional method for adversaries to confirm that an email address does or does not exist. Additionally it allows for a new channel for materials to be delivered to the recipient’s mail processor that is not checked for malware or viruses by the standard mail scanning methods in place. For these reasons recipient processing systems need to implement the following counter-measures:

1) The pointer to the PDEP MUST be checked against some policy before attempting to query the PDEP for a policy decision. 2) Care MUST be taken when processing the responses from a PDEP check that they are well-formed and meet local policy before using the responses.
Editorial Comments
Appendix A. References

A.1. Normative References


A.2. Informative References

[bc-iaf] Province of British Columbia; Electronic Credential And Authentication Standard, version 1.0
[lib- iaf] Liberty Alliance; Liberty Identity Assurance Framework, version 1.1
[SAML-over] OASIS, Security Assertion Markup Language (SAML) Version 2.0 Technical Overview
Appendix B Authors’ Addresses

Trevor Freeman
Microsoft Corp.
Email: trevorf@microsoft.com

Jim Schaad
Soaring Hawk Consulting
Email: ietf@augustcellars.com

Patrick Patterson
Carillon Information Security Inc
Email: ppatterson@carillon.ca
Appendix C Document Change History

Fixed comments on 07 draft from document shepard