Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 6, 2015

T. Freeman
J. Schaad
Soaring Hawk Consulting
P. Patterson
Carillon Information Security Inc

March 5, 2015

Requirements for Message Access Control
draft-freeman-plasma-requirements-11

Abstract

S/MIME delivers confidentiality, integrity, and data origination authentication for email. However, there are many situations where organizations also want robust access control applied to information in messages. The Enhanced Security Services (ESS) RFC5035 for S/MIME defines an access control mechanism for email, but the access check happens after the data is decrypted by the recipient which devalues the protection afforded by the cryptography and provides very weak guarantees of policy compliance. Another major issues for S/MIME is its dependency on a single type of identity credential, an X.509 certificate. Many users on the Internet today do not have X.509 certificates and therefore cannot use S/MIME. Furthermore, the requirement to discover the X.509 certificate for every recipient of an encrypted message by the sender has proven to be an unreliable process for a number of reasons.

This document presents requirements for an alternative model to ESS to address the identified issues with access control in order to deliver more robust compliance for S/MIME protected messages. This document describes an access control model which uses cryptographic keys to enforce access control policy decisions where the policy check is performed prior to the decryption of the message contents. This authorization model can be instantiated using many existing standards and is in not intended to be a one off just for email, being applicable to other data types.

This document also presents requirements for the abstraction of the specifics of the authentication technologies used by S/MIME users. The abstraction makes it possible for other forms of authentication credentials to be used with S/MIME thereby enabling much broader adoption. The authentication abstraction model also removes the dependency on the need to discover encryption keys by the sender. This abstraction can be used independently from access control to enable simple scenarios where authentication of the recipient is sufficient to grant access to the message.

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

Copyright Notice

Table of Contents

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.

1 Policy-Based Management Vocabulary

This document uses the established terminology for policy-based management [RFC3198] where applicable. The following list supplements the terms defined in [RFC3198] as well as defining some new combinations of terms used in [RFC3198].

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.

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 through 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.

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.

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.

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.)

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 and Enforcement Point (PDEP)

The system entity that both evaluates the policy criteria published by a PAP, using attributes supplied by a PIP to render decisions on requests made by DRs and enforces its decision via the use

of cryptographic keys.

Policy Decision Point (PDP)
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 PDP has a separate enforcement point.

Policy Enforcement Point (PEP)
The system entity that enforces the decisions of a 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.

Policy Information Point (PIP)
A service which issues assertions, for example about a subject, their device, or environment e.g., an LDAP directory or SAML Security Token Service.

Policy Label
The data structure which holds one or more policy identifiers and their logical relationship.

Pseudonym
A name that a person or group assumes for a particular purpose, which differs from their original or true name. (See Orthonym.)

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.

2 Introduction

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 key exchange mechanisms to be used, S/MIME [RFC5750] exclusively uses X.509 certificates [RFC5280] for the security credentials for signing and

encryption operations. S/MIME also uses an early binding mechanism for encryption keys where the sender needs to discover the public key for every 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 and is a major factor in the lack of adoption of S/MIME. The S/MIME key management model has many dependencies resulting in senders often unable to encrypted email to recipients. For example, to encrypt a message, the sender needs to discover the X.509 certificate for every recipient. This may not be possible for a variety of reasons:

- The recipient may not have an X.509 encryption certificate
- The sender may not have previously received a signed email with the recipient's certificate
- The recipient may not have an available repository from which to publish their certificate for senders to discover
- 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 valid certificates from the message.

The use of secure mailing lists has the ability to provide some relief to the above problems, especially for cross-domain scenarios. The original sender only needs to know the appropriate encryption information for the mailing list in the other domain; the mailing list in turn, enables its recipients to decrypt the message as part of the mailing list expansion. It can thus be thought of as a form of late-binding of recipient information for the originating sender. As a solution it therefore helps where all recipients have trusted certificates but the certificates of recipients in other domain are not discoverable by the sender in their domain. The certificate of the mailing list is however discoverable by the sender, and it enables all email sent to the mailing list to be encrypted to named recipients on the mailing list in all domains.

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 should be granted. In many inter-organization collaboration scenarios

it's impossible for the sender to satisfy the access checks on behalf of recipients in other organizations since that would require the sender's client or an agent in the sender's domain to access to those recipients' attributes to perform the access check, which may be a breach of the recipients' privacy. Indeed to release the attributes to the sender may require that the sender's attributes first be released to the recipients' attributes provider to authorize the release of the recipients' attributes. It's also a fundamental tenet of good privacy practice that consent should be obtained from users before release of data about themselves.

ESS Security labels are an optional security service for S/MIME. 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 policy. 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 requirements 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.

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. While labels cannot be altered, they 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 an 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 users 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 robust 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. The biggest issue with ESS labels is 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 enforced 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 recipient's machine will have the updated module. This makes policy compliance practically impossible in anything but a small, closed environment.

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.

S/MIME today can only use X.509 certificates to protect the

confidentiality or the data origin authentication and integrity of the messages. There are many users on the Internet today who have other forms of authentication credentials. This means the many users without X.509 certificates cannot use S/MIME. There have been many developments in authentication technology and best practices since S/MIME was developed over a decade ago, and example of which is SAML [SAML-core]. The critical difference between SAML and X.509 certificates is that SAML abstracts the details of the authentication protocol from the application protocol. The identity provider can use a broad range of authentication mechanisms via SAML such as passwords, one-time passwords, biometrics, X.509 certificates, etc., to authenticate the subject without impacting the relying part or application protocol. Adopting the abstraction model for S/MIME would enable almost anybody with any kind of authentication credential registered with one of the many identity providers on the Internet today to use S/MIME making it possible that S/MIME use may become as pervasive as TLS is today.

There are many other non-email use cases which would be subject to the same access policy requirements. Email allows users to create content and distribute it to a set of recipients. Similar use cases can be performed with other data formats or applications such as documents and instant messages. Policy is tied to the information, not the data format or application, therefore if an organization has a policy relating to a type of information, then that same policy would apply to the same information in any form; email, document, or instant message. While some aspects of this work will be specific to email, there will be many which would be reusable in other areas.

3. Access Control Models

Access control is the process whereby systems are able to decide whether to grant a request to access a resource from a subject. There are a number of models the system can follow to make the decision. These are two types of models, those based on a subject attributes and those based on a subjects capabilities. For models based on subject attributes, the system obtains a set of attributes about the subject then applies a policy expression using the attributes as input to the policy to determine the result. For models based on subject capabilities, the subject has an unforgeable token or reference to a token attesting to an access to a resource.

The simplest model based on subject attributes is Discretionary Access Control (DAC) where subject attributes are the subject's identity and their group memberships. The access control policy is expressed as an Access Control List (ACL) which is a list of identities and or groups together with the allowed (or denied) access for each entry. The ACL is evaluated sequentially, and the first match is the access control

decision. Under the newer taxonomy for access control models, DAC is Identity based Access Control (IBAC) where the access control is based on the subjects identity or the identity of a group they belong to.

Role Based Access Control (RBAC) is a refinement of DAC where the role is an abstract identity associated with a specific function which is granted a set of permissions and a subject can be assigned one or more roles. When a subject is assigned a new function, they are assigned the role for that function. The role used to simplify management, in essence each role is a collection of groups.

Capability Based Access Control (CBAC) which is based on unforgeable tokens which contain a reference to an object together with the access permissions to the object. CBAC token were initially implemented as privileged data structures by the operating system, where possession of or reference to the token grants the bearer the access rights defined in the token. Modern cryptographic techniques allow the tokens to be integrity protected so can be passed more openly between systems.

Attribute Based Access Control (ABAC), where policies are defined in terms of arbitrary attributes of the subject, their device or environment, their intended action on or use of the information. ABAC requires the definition of the policy in a policy expression language, e.g., eXtensible Access Control Markup Language [XACML-core]. ABAC also requires a secure way to exchange arbitrary attributes, e.g., via the Security Assertion Markup Language [SAML-core] or via an LDAP directory.

SAML [SAML-core] defines an XML framework for describing and exchanging assertion tokens containing attributes. The entity issuing the assertion tokens is a Policy Information Point. The entity consuming the assertion with the attributes is known as the relying party (RP). The well-known scenarios for using SAML are:

o Single Sign-On across systems on different platform technology

o Federated Identity between business partners

o Web Services and other standards, e.g., SOAP-based protocols

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 relying party together with the token being issued for use with the RP. 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 certificates the subject proves its identity to the RP by demonstrating control over the key, e.g., signature or HMAC over some data. The RP can therefore have a stronger proof of identity by the demonstration of possession of cryptographic keys. SAML can also be used to express attributes about a subject to an RP where the subject has authenticated to the RP by some means.

To prevent every relying party from having to become familiar with the specifics of the various types of subject identity proofing, authentication technologies etc., used by identity providers (IdP), abstraction frameworks have been developed which are taxonomies used by the IdPs to classify the level the level of assurance of the subject identity. The taxonomies allow the resultant LoA of the identity to be represented by a number (1 to n) where 1 is the lowest level of assurance and n is the highest defined by the framework. The framework provides 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 by subjects to employ tokens to authenticate to an identity provider

- Protocols used by subjects to authenticate and pass attributes to a relying party

The relying party simply has to determine the LoA required for access as defined by the framework. The framework ensure consistent evaluation of the LoA of the identity for the relying party. It also means the IdP can change some of the specifics e.g. deploy a new authentication technology, without impacting the relying part of their policies. These framework have been drafted by industry organizations and governments. 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. A simplified description of the levels is:

Level of Assurance	Confidence in the Asserted Identity
Level 1	Negligible Confidence
Level 2	Some Confidence
Level 3	Significant Confidence
Level 4	High Confidence

3.1 Generic Access Control Model

The terminology defined in [RFC3198] uses a generic information model for the actors and the way they relate to each other. This work extends the generic model in the RFC to accommodate ABAC.

```
                                       ------------------
                                       |                |
                                       |     Policy     |
                                       | Administration |
                                       |     Point      |
                                       |                |
                                       ------------------
    -----------------                          |
    |               |                          |
    |   Policy      |                          |  Read
    |  Information  |                          |  Policy
    |   Point       |                          |
    |               |                          |
    -----------------                          v
         |  |                                  v
         |  |                          -----------------
         |  |   Back-end Attribute     |               |
         |  |   Exchange               |     Policy    |
         |  ------------------------->>|    Decision   |
         |  Issue                      |     Point     |
         |  Attributes                 |               |
         |                             -----------------
         |                                     ^
         |  Front-end Attribute                ^  Decision
         |  Exchange                           |  Request +
         v                                     |  Attributes
         v                                     |
     -----------------                 -----------------
     |               |    Request +    |               |
     |  Subject      |    Attributes   |    Policy     |
     |  Decision     | --------------->>|  Enforcement  |
     |  Requestor    |                 |    Point      |
     |               |                 |               |
     -----------------                 -----------------
```

Figure 1 Generic Access Control Model

o Administrators manage and publish policies using the PAP. The published policies are then available to the PDP

o A decision requestor sends a request together with its attributes to the PEP

o The PEP sends a decision request to the PDP together with the subject attributes

o The PDP obtains the necessary policy from the PAP

o The PDP can request additional attributes from the PIP

o The PDP returns the decision responce to the PEP

o The PEP enforces the decision

This generic model assumes the PEP has control over the data i.e. when it gets the grant decision, it releases the data to the subject. This works well in client-server situations like access to a web site or database where there is a clear trust boundary between the subject and the PEP with the data. However it does not work well with applications like email where the data is delivered to the subject prior to the access check. The model needs to be extended to allow the data to be encrypted and the access check be performed prior to release of the decryption key.

A dependency in the model is the reliability of the policy selection for the request by the PDP. The implementation of the policy selection process can make either a closed- or open-world assumption. Closed-world assumes the policy set on the PDP is complete therefore there is a policy in the store for every request. Open-world assumes the policy store is incomplete and there is a need to discover new policies as appropriate. Closed-world implementations work when there is reasonable control over the sets of data managed by the PEP and policies known to the PDP. However they result in unreliable results with mobile data, i.e., if data is received from a partner and an attempt is made to process it via the recipient's PEP and PDP. There is no linkage between the distribution of the data and the distribution of the policies in closed-world models. It is therefore possible that data will be received for which the matching policy is not available from the recipients policy store.

Access control models based on subject attributes depend upon the availability of assertions with attributes about subjects. The model has the PIP issuing attributes about subjects and the PDP consuming attributes about subjects. A subject can be a human, a device, or a service. The subject must have a relationship with the PIP since it has been through some form of registration process with the PIP. There is no requirement to have a relationship between the PIP and a PDP. The PDP must trust the PIP, but not vice versa. This is the same model as exists with X.509. The subject must have a relationship with the CA, the RP 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 RP. Release of subject attributes to a PDP must be under a policy due to the sensitivity of the data. The subjects themselves can request and give approval for the release of attributes from the PIP and relay them to the PDP (Front-end Attribute Exchange). If the subject has given prior consent, the RP may receive attributes directly from the PIP(Back-end Attribute Exchange). Subject attributes are potentially sensitive data and are similarly subject to access control. SAML has the capability to encrypt sensitive data in the token. The PIP would also develop policy to regulate the set of data it would release to a PDP.

The challenges for S/MIME are therefore:

- How to apply this generic access control model to the email scenarios so there is convergence with other applications, i.e., email access control is not a one-off, vertical solution

- How to ensure the access check is possible prior to the recipient having access to the clear text so the access check is sufficiently robust for regulators

- How to abstract the authentication credential technology use from the S/MIME protocol to enable use of the many forms of authentication in widespread use today on the Internet.

4 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.

4.1 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 issue 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 identity 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 the message.

4. Alice's email client sends the recipient list and encryption key to a server and in return gets a message token.
5. Alice's client attaches the token to the message and sends the message which is able to flow securely and seamlessly through existing email infrastructure to Bob. The data is protected while in transit and at rest.
6. Bob receives the email and sees that it is a secure message. Bob's email client uses the message token to verify that the secure message has not been altered.
7. Bob's email client attempts to open and decrypt the email using the message token. If Bob is on the same ISP as Alice, then the same username/password as he uses to get his email is likly 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 authorize the needed decryption keys to be released.

4.2 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. This example is illustrative of the many use cases for business-to-consumer email.

A bank (The Bank of Foo) has determined that it will be using email to distribute statements to its customer (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 i.e. LoA=>2. The same identity assurance authentication level used to do on-line 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 recipient's 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 on-line 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 business or organization 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. Bob is able to use the same credential when sending or replying to a message from the bank, as he uses for accessing the bank's Web site then the bank has the same assurance of Bob's identity for all transactions.

4.3 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., for contract negotiations. This level of security is similar to guarantees to the security afforded by mail, i.e., you enclose a letter in an envelope which provides a level of security to the contents while in transit. There is an expectation that only the recipient or their delegate would open the envelope. Once the recipient has the letter, you trust them to treat the contents appropriately.

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. Charlie trusts Dave to treat the information appropriately. When Charlie sends the email to Dave with the sensitive content, he must ensure the following objectives:

(a) Only Dave or his delegate can read the email.
(b) Dave or his delegate is required to authenticate with a LoA => 2
(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 include 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 the 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's policy requires LoA 2 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.

4.4 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 messages 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 (years). The set of policies applicable to an email is potentially subject to change as the different user's contribute information to the email thread.

4.4.1 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 sets 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 is 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 all applicable policies based on the message contents. 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 all necessary policies 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. Because she does not know all the details of the policies, Grace would not know the set of attributes she needs to disclose to the PDEP to access the message. Grace starts with a basic set of attributes to identify herself. The PDEP may be able to discover more attributes about Grace once it knows her identity, and if it unable to find all it

needs, it can request any missing attributes from Grace.

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) All recipients meet the necessary policy requirements based on the message contents; if the message contains Program X export control data, they meet that policy; if it contains Company Foo's intellectual property data, they meet that policy; if it contains both types of data, then recipients must meet both policies.
(b) Recipients authenticate with an identity assurance as required by the policies, e.g., LoA 3 or above.
(c) Recipients present all other attributes about themselves necessary to verify compliance with the applicable policies (e.g. 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 which is export controlled. 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 export control 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, based on the policies selected, that the message needs to be signed with a LoA 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 a token for the message from a server that will enable the recipients' servers to enforce the access control requirements for Frank, and sends the email 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 Content Encryption Key (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.
(9) The CEK Grace received has a Time To Live (TTL) value which defines when Grace must discard the CEK and reapply for a new CEK.

Grace is able to open the message and can reopen the message without reauthorization within the time window of the TTL. Once the TTL expires, she must discard the CEK. If she needs subsequent access, she must reauthorize her access to the message in the same way as when it first arrived. If some of her attributes change, e.g., Grace is removed from Program X by Company Bar, then she may not get access to the message in the future.

If Grace replies to the email from Frank, because Company Bar has decided agreed to enforce the Company Foo policies for protecting Company Foo data, 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 wants consistent policy enforcement on their information and have chosen to use each other policies as a means to

protect each other's data.

After some time, Company Bar fails an audit to show they are complying with 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. All company Bar employees will then no longer be able to receive new CEKs for Program X emails as they can no longer satisfy the Program X policy requirements. Existing CEKs should be discarded after the expiration of the TTL.

4.4.2 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 Program X (e.g., when a design is complete and released to manufacturing). The sign-off process requires personnel be suitability qualified and that the documentation needs to be maintained for the service life of the project (25 years for Program X).

Company Foo has instigated an email-based sign-off procedure to simplify sign-off and reduce costs. At the appropriate time, a sign-off request policy email is sent to the designated program members. If recipients want to approve, they reply to the request using the sign-off-approved policy as required by the sign-off request policy.

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 request email. Grace responds and applies the sign-off approved policy to the email. The policy requires Grace to authenticate with the required level of identity assurance, present attributes about herself, her device, her work effort assignments and professional qualifications, and attributes about the data being approved to demonstrate compliance with the policy. The sign-off approved policy has some options for how to communicate attributes about the data being approved. It can require Grace to send a hash of the data over a secure transport, or she can send a signed hash of the data if she has a suitable signing certificate, she can send the data itself. The policy will define what options are acceptable.

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 have not been modified to the level of assurance required by the policy.
(d) Frank was made aware that is this a sign-off request email and confirms he 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 of the approval message can confirm over the lifetime of the program, that the signature was applied because the sender of the approval message complied with the approval policy at the time of the signature. The confirmation of the policy approval does not require knowledge the specifics of the policy.

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.
(5) Grace's system submits the decision request to send the sign-off email. Her system is asked to provide attributes about Grace, the state of her system and the data being authenticated as part of the decision request.

Grace would not know the complete set of attributes required to submit her sign-off as she does not know what the policy requires and would start with a basic set to identify herself. The PDEP may be able to discover additional attributes about Grace, and if it is still missing some, can request those missing attributes from Grace. If Grace's request meets the policy, her system receives a signed statement from the PDEP that the message meets the policy which is attached to the email and the message is sent.

4.5 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.

4.6 Policy Compliance Verification

Verification is an essential part of compliance. Verification of compliance with a policy may be conducted by internal staff or external auditors. The verification need to confirm that the policy rules are being enforced, e.g., when data is accessed. 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 a distributed system and data, 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 policies 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.

4.7 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 the MTA for this policy, and if it does, that Company Foo has an outbound email scanning capability, 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.

4.8 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 policies may have the list of recipients as a parameter of the policy in the message, thus the fact that the message is being processed by the distribution list means the MTA processing the message needs to expand the list of recipients 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.

4.9 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.

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 PDEPs in both Company Foo and Company Bar. Company Foo has also decided to use an export-controlled approved 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.

5 Plasma 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 store such data. The binding between the data and the applicable policies 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 2 shows the relationship between the actors.

```
                            ------------------
                            |                |
                            |     Policy     |
                            | Administration |
                            |     Point      |
                            |                |
                            ------------------
                                     |
    -----------------                |               -----------------
    |               |                |               |               |
    |   Policy      |                |  Read         |   Policy      |
    |  Information  |                |  Policy       |  Information  |
    |   Point       |                |               |   Point       |
    |               |                |               |               |
    -----------------                v               -----------------
         |  |                        v                       |  |
         |  |Issue          -----------------      Issue     |  |
         |  |Attributes     |               |      Attributes|  |
         |  |(BAE)          |     Policy    |      (BAE)     |  |
         |  -------------->>|    Decision   |<<---------------  |
         |                  |      and      |                   |
         |                  |  Enforcement  |                   |
         |  -------------->>|     Point     |<<-----------      |
         |  |Protect        |               |  Consume   |      |
         |  |Content        -----------------  Content   |      |
         |  |Request+                          Request+  |      |
         |  |Attributes                        Attributes|      |
         |  |(FAE)                             (FAE)     |      |
         v  |                                            v      v
         v  |                                            v      v
     -----------------                               -----------------
     |               |                               |               |
     |   Content     |           Distribute          |   Content     |
     |  Creation     |           Content             |  Consumption  |
     |  Decision     | ---------------------------->>|  Decision     |
     |  Requestor    |                               |  Requestor    |
     |               |                               |               |
     -----------------                               -----------------
```

Figure 2 General Scheme for Publishing and Consuming Protected Content

The Plasma model is applicable to any type data (email, documents, databases, IM, VoIP, etc.). This facilitates 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 the content. The content consuming DR dynamically discovers the PDEPs that are authoritative for the decisions on protected content in question. The PDEPs 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.

o   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 enforced by the PDEP is bound to the protected data.

O   The Plasma model includes policy discovery capability for subjects. This enables subjects to interact with one or more PDEPs to discover the set of policies each PDEP would permit the subject to use to protect new content. The PDEP issues a plasma 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 policies from a policy collection, but cannot mix policies from different policy collections.

o   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.

o   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.

o   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.

o   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.

o   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 its 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 Identity Providers(s) able to issue identity attributes to the subject
- A means to authenticate to the Identity Providers(s)and issue attributes to the subject
- The address of zero or more Attribute Providers(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 Identity Provider 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-organizational collaboration. The PDEPs 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 a signed data structure created by the PDEP which is bound to the protected data, i.e., it has a detached signature over the encrypted 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 example 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 policies. The tree has a series of nodes where each node represents a set of policies and the relationship for the policies at the node, e.g., are they combined via and AND clause or an OR clause. The pinnacle of the tree represents the decision from all the policies in the tree. The use of policy references minimizes any policy maintenance issues relating to the protected data due to policy updates. The policy rues 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 device 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.

```
  ---------------          -----------------         -----------------
  |             |          |               |         |               |
  |             |          |   Policy      |         |  Policy       |
  |  Policy     |          |  Decision and |         |  Decision and |
  | Decision    |          |  Enforcement  |         |  Enforcement  |
  |  Point      |          |   Point       |         |  Point        |
  |             |          |               |         |               |
  ---------------          -----------------         -----------------
         |                         |                         |
         |                  T      |      T                  |
         |                  TTTTTTT|TTTTTTT                  |
         V                         V                         V
         V                         V                         V
  ---------------           ---------------           ---------------
  |             |           |             |           |             |
  |  Policy     |           | Decision    |           | Decision    |
  | Enforcement |           | Requestor   |           | Requestor   |
  |  Point      |           |             |           |             |
  |             |           |             |           |             |
  ---------------           ---------------           ---------------
         |                         |                         |
  T      |     T                   |                         |
  TTTTTTT|TTTTTT                   |                         |
         V                         V                         V
         V                         V                         V
  ---------------           ---------------           ---------------
  |             |           |             |           |             |
  |  End        |           |  End        |           |  End        |
  |  User       |           |  User       |           |  User       |
  | Application |           | Application |           | Application |
  |             |           |             |           |             |
  ---------------           ---------------           ---------------
        (a)                        (b)                      (c)
```

Figure 3 Options For Trusted Actors with Data.

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 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.

5.1 Plasma Client/Server Key Exchange Level of Assurance

There are a number of mechanisms by which a client and server 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.

5.2 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 a decision request is made for access 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 reference is that once the reference is bound to the data, the policies continue to be consistently applied over time over multiple instances of the data and as the policies change over time. Another strength of binding policy by reference is

it has a clear result as to the set of policies the PDEP has to apply. If the PDEP 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 reference 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 a 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 that needs to be captured and the policies that need to be enforced. 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 PDEP having a complete the set of policies (closed-world assumption). 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 different PDEPs, it's impossible to determine the correctness of the result of the policy matching process by the new PDEP.

    The Plasma model is choosing to use binding by reference 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.

5.3 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 to each PDEP and the contents of the plasma role token authorized by each 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 PDEPs. 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 for the duration of the tokens TTL 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 service decision requests for the data (the content creation PDEP does not have to be in the set of PDEPs allowed to 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.

5.4 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 the use of the data.

(d) The DR sends the read token request to one of the URL's 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 needs 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.

5.5 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 PDEP 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.

```
          Internet |         DMZ             |       Intranet
                   |                         |
                   |                         |
                   |                         |      ---------------
                   |                         |      |             |
             TLS   |                         | TLS  |  DR         |
         ----------|<------------------------|------|  Client     |
                   |                         |      |             |
            (a)    |                         |      ---------------
          no proxy |                         |
                   |                         |
                   |                         |
                   |      ---------------    |      ---------------
                   |      |             |    |      |             |
             TLS   |      |  Plasma     |    | TLS  |  DR         |
         ----------|<-----|  Forward    |<---|------|  Client     |
                   |      |  Proxy      |    |      |             |
             (b)   |      |             |    |      ---------------
           Forward |      ---------------    |
           Proxy   |                         |
```

Figure 4 Plasma Forward Proxy

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.

```
          Internet |         DMZ             |       Intranet
                   |                         |
                   |                         |
                   |      ---------------    |      ---------------
                   |      |             |    |      |             |
             TLS   |      |  Passive    |    | TLS  |  Full       |
         ----------|------|-------------|----|----->|  PDEP       |
                   |      |  Reverse    |    |      |  Server     |
                   |      |  Proxy      |    |      |             |
            (a)    |      |             |    |      |             |
                   |      ---------------    |      |  TLS Keys,  |
                   |                         |      |  Message    |
                   |                         |      |  Encryption |
                   |                         |      |  Keys       |
                   |                         |      |             |
                   |                         |      ---------------
                   |                         |
                   |      ---------------    |      ---------------
                   |      |             |    |      |             |
             TLS   |      |  Active     |    | TLS  |  Full       |
         ----------|----->|  Reverse    |----|----->|  PDEP       |
                   |      |  Proxy      |    |      |  Server     |
             (b)   |      |             |    |      |             |
                   |      |  TLS keys   |    |      |  TLS Keys,  |
                   |      |             |    |      |  Message    |
                   |      ---------------    |      |  Encryption |
                   |                         |      |  Keys       |
                   |                         |      |             |
                   |                         |      ---------------
                   |                         |
                   |                         |
```

Figure 5 Plasma Reverse Proxy

5.6 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.

5.6.1 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 attributes required which will simplify development, testing, and deployment. Later standards may expand the set of attributes supported by basic policies and hence define richer basic policies.

5.6.2 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 a PIP. 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.

6 Message Protection Requirements

6.1 General Requirements

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

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

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 SHOULD be abstracted from the PDEP and DR by use of a simple numeric scale linked to an identity assurance framework that defines the specifics of how to derive the LoA. (See sections 4.1, 4.2, 4.3 and 4.4.)

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 message so the policy changes can be

implemented without updating the message. (See section 4.4.)

Access to the plaintext of the message MUST only be provided after the recipient has either provided suitable valid attributes to the PDEP or the PDEP finds attributes about the recipient directly from a PIP, thus satisfying the policy as defined by the PAP. (See sections 4.1, 4.2, 4.3, 4.4 and 4.5.)

The sender MUST be able to obtain a list of policies to messages they create and scoped to their current role(s), i.e., what tasks they are currently assigned to deliver. (See sections 4.1, 4.2, 4.3 and 4.4.)

The specifics of the access control policy MUST be abstracted from both the sender's and the recipient's, i.e., they MUST NOT make the access control decision or need specifics of the access policy requirements. (See sections 4.1, 4.2, 4.3 and 4.4.)

A recipient MUST receive authenticated attributes of the identity and level of identity assurance of the sender. (See sections 4.1, 4.2, 4.3 and 4.4.)

The key exchange between sender, recipient, and the PDEP MUST support multiple levels of assurance of the exchange. 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 is required to provide a suitable key exchange key such as an X.509 certificate to encrypt the CEK. (See sections 4.3 and 4.4.)

The level of key exchange assurance required MUST be selected by the sender's policy and enforced by the PDEP. (See sections 4.1, 4.2, 4.3 and 4.4.1.)

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

A time-to-live (TTL) MUST be provided to recipients when access is granted to a message by the PDEP to define when the recipient 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 recipient and its environment. (See section 4.4.1.)

The PDEP MUST be stateless for processing policy requests from senders and recipients with respect to any instance of a message. 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. (See section 4.9.)

A PDEP MUST be capable of generating audit events associated with access to protected content using policy defined by the PAP. (See section 4.6.)

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. (See sections 4.7 and 4.8)

It MUST be possible for MTAs to request access to protected messages for which they have not been authorized by the sender. (See section 4.7 and 4.8.)

It SHOULD be possible for an MTA to pre-authorize another to access a protected message. (See section 4.7 and 4.8.)

6.2 Basic Policy Requirements

The use of a Basic policy MUST be backwards compatible with existing S/MIME, i.e., it MUST be possible to both to exist on the same message.

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 the definition of other Basic policies to be defined by other groups, trust frameworks, or even within the context of the IETF.

When using a Basic policy defined by this work, the sender MUST define which Basic policy is required and the list of [RFC5322] recipients.

The list of recipients MUST be able to be appended to by a distribution list expansion server.

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

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

message.

6.3 Advanced Policy Requirements

A Basic policy MAY be combined with Advanced policies on a message

The use of an advanced policy and existing S/MIME confidentiality on the same message MUST NOT be permitted.

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

Where two or more policies are applied to a message, the logical relationship between the policies MUST also be expressed, e.g., are the policies a logical AND or a logical OR. (See section 4.4.)

An Advanced policy MAY require attributes about:

- The recipient
- The recipient's device
- The environment of the device that is attempting to access the message.
- The message being accessed

Advanced policy MUST support an extensible list of obligations on the sender or recipient or PDEP such as use of the policy requires some specific action on the part of the sender, e.g., signing content with a 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.

7 IANA Considerations

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

8 Security Considerations

Authentication by itself is not a good trust indicator. Authentication raises the level of assurance that 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 on 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. Domains should have sufficient granularity of policy to handle situations where their email pipeline agents are not able 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.

Email is frequently used as part of a password reset ceremony by an identity provider. This is problematic when combined with access to sensitive email. This could be part of an escalation attack, e.g., compromise low value email account password, initiate password reset via email for higher value account. This would then give access to any email protected using the higher value identity.

Providing differential access to different parts of a message based on different policies should only be done via use of different encryption keys. All data protected by the same key is under the same access control policy.

It would be desirable to be able to indicate the times and other data like request location when a user has asked for access (successful or otherwise) to some content as a means to show malicious activity to the user.

Part of the policy is obligations on how to protect the data, e.g., algorithms and parameters required. This can change over time,

therefore a client may become obligated to re-encrypt or re-digest the data if it encounters data which does not meet the current mandate.

The act of requesting access to messages is a potential privacy issue as it allows the sender to gather data about the recipient. For business-to-business transactions, disclosure of employee information is handled by the organization. For consumers, there is a need to be able to consent to the privacy obligations associated with disclosure of information. This would include information the consumer releases to the PDEP as well as information the PDEP is able to gather such as time and location of access requests.

The fact the PDEP is able to grant access to the data could be used by law enforcement to access information. One of the parameters the sender needs to be aware of is the jurisdiction the PDEP is under so they can make an informed choice.

Appendix A. References

A.1. Normative References

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3198]   Westerinen et. al., "Terminology for Policy-Based Management", November 2001.

[RFC5035]   Schaad, J., "Enhanced Security Services (ESS) Update", August 2007.

[RFC5280]   Cooper, D, et al, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC5280, May 2008

[RFC5322]   Resnick, P., "Internet Message Format", RFC5322, October 2008.

[RFC5652]   Housley, R., "Cryptographic Message Syntax (CMS)", RFC 5652, September 2009.

[RFC5750]   Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Certificate Handling", RFC 5750, January 2010.

[RFC5751]   Ramsdell B., Turner S., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", January 2010

[SAML-core]   OASIS, Assertions and Protocols for the Security Assertion Markup Language (SAML) Version 2.0, March 2005

A.2. Informative References

[RFC3114]   Nicolls, W., "Implementing Company Classification Policy

with the S/MIME Security Label", RFC 3114, May 2002.

[RFC5408]   Appenzeller, G., "Identity-Based Encryption Architecture and Supporting Data Structures", RFC5408, January 2009.

[XACML-core] OASIS, eXtensible Access Control Markup Language (XACML) Version 3.0 Core Specification

Appendix B Authors' Addresses

Trevor Freeman

Email: trevor.freeman99@icloud.com

Jim Schaad

Soaring Hawk Consulting

Email: ietf@augustcellars.com

Patrick Patterson

Carillon Information Security Inc.

Email: ppatterson@carillon.ca