Content-based Protection and Release Architecture for Future NATO Networks

Konrad Wrona
NATO Communications and Information Agency
The Hague, Netherlands
konrad.wrona@ncia.nato.int

Sander Oudkerk
Agent Sierra Consultancy Services
Amsterdam, Netherlands
sander.oudkerk@agentsierra.nl

***Abstract*— Future mission networks will require agility and flexibility in sharing information between heterogeneous users and terminals. We introduce the Content-based Protection and Release (CPR) model and present its applicability to current and future information sharing use cases in NATO. We present a service-oriented CPR architecture, which was developed at the NATO Communications and Information (NCI) Agency in order to address the evolving information sharing requirements in NATO operations. The paper highlights a number of important security services, which together comprise the CPR architecture. The central component is the CPR Enforcement Separation Service (CPRESS). CPRESS is supported by several other security services, such as the NATO Metadata Binding Service and the Content Inspection Policy Enforcement service. We describe these services and the interfaces between them. We also include recommendations based on experimentation with a CPR demonstrator developed at the NCI Agency, discuss related work, and identify possible future directions for research and development activities.**

Keywords—Attribute-based access control; labelling; guards.

I. INTRODUCTION

The concept of Content-based Protection and Release (CPR) [1] was developed at the NATO Communications and Information (NCI) Agency as an application of Attribute-Based Access Control (ABAC) [2]. The aim of CPR is to improve timely sharing of information in the NATO Network-Enabled Capability (NNEC) and Future Mission Network (FMN) environments [3]. In current operations timely sharing of information is hampered by a number of limitations that are inherent to the traditional use of sensitivity labels. CPR overcomes these limitations by enforcing access control based on a content label, which contains content properties of an information object, instead of based on a sensitivity label.

II. LIMITATIONS OF THE USE OF SENSITIVITY LABELS

The values of a sensitivity label are determined by the information creator based on the (information) security policy. The sensitivity label reflects the protection requirements and release conditions of an information object at the time of creation and does not capture the reasons for the protection requirements and release conditions. This means that when circumstances give rise to a change in the protection requirements and release conditions of an information object, such a change can be effected only by re-labelling the information object.

The process of re-labelling an information object can be time-consuming and must be executed following strict information management procedures initially involving consultation with the information object originator. If the originator cannot be determined or is otherwise unavailable it becomes necessary to execute a manual review of the information. The inefficiency of the process of re-labelling prevents a prompt response to a change in the handling requirements of information and therefore hampers timely sharing.

In addition to lack of timeliness in information sharing, information flow may become inefficient as a result of the use of incorrect or imprecise sensitivity labels. A sensitivity label may be incorrect as a result of misinterpretation of the security policy based on which the values of a sensitivity label must be determined. A common negative consequence is under- or over-classification of an information object. A sensitivity label may be imprecise in the sense that it is incomplete, e.g. it lacks the required release categories because such categories were not deemed relevant by the information object originator at the time of creation of the information object. Note that there is an inherent limit to the precision of a sensitivity label: the complexity of the dissemination control policy that can be expressed within the label is bound by the (relatively simple) label syntax and its values.

Lastly, situational awareness and access to information in general can be negatively influenced by subjective interpretation of the security policy by information originators. Subjective interpretation of the security policy may lead to the derivation of different sensitivity labels for information objects with similar content. As a result similar information is protected in different ways and may therefore not be equally accessible.

III. CONTENT-BASED PROTECTION AND RELEASE

In CPR access to an information object O does not depend on a sensitivity label but instead on a content label. The content label is a structured representation of a set of content properties S that belong to O. The decision to provide a user access to O is based on separately managed protection and release policies that, for the set S, express the requirements the user and the operational environment (e.g. user's terminal) must meet in

978-0-7695-5124-1/13 $31.00 © 2013 Crown Copyright
DOI 10.1109/MILCOM.2013.44

order to access *O*. The requirements are translated to user attributes and, in the case of the user's terminal, to terminal attributes. In the remainder of this paper CPR is described for information requests made by a user from a terminal.

CPR removes the disadvantages of the use of sensitivity labels because it separates the labelling of information objects from the process of determining the protection requirements and release conditions. In CPR the information object originators are no longer required to interpret the current security policy and derive the values of a sensitivity label. Instead they have to describe the object by assigning content properties (which are captured in a (structured) content label and bound to the information object). The selection of content properties is less susceptible to subjective interpretation and hence increases the likelihood of the (content) label being correct; it is assumed that information object creators are qualified to correctly select content properties, given that they know the content of the information object.

Content labels can also be made much more precise than traditional sensitivity labels (in the way these are currently used in NATO) by adding more-detailed content properties. Adding more-detailed content properties provides the additional advantage that more precise protection requirements and release conditions can be derived. When two information objects with similar content have different originators, the use of content properties will increase the likelihood that the protection requirements and release conditions of both objects will not deviate significantly. The use of standardized content property catalogues further increases this likelihood. The use of such catalogues will also guarantee correct correspondence between the content properties assigned to information objects and the content properties in terms of which the protection and release policies are expressed.

The policy management authority of the information domain in which the information object is created must maintain the protection and release policy. The protection and release policies can be made as detailed as required and are dynamic in the sense that they can be updated at any time if circumstances so dictate. For this reason, the actual protection requirements and release conditions of an information object are derived at the moment an access request for the information object is made. Note that if there is a reason to change the protection requirements or release conditions of an information object *O*, it is no longer required to change the label of *O* since the content properties of an information object remain the same. Instead the protection and release policy will be updated. Thus, a change in protection requirements or release conditions is reflected immediately without the need for manual review or consultation with the originator.

Although in CPR sensitivity labels are no longer used in the process of deriving and enforcing an access control decision, it is still necessary to inform human users of the release conditions and protection requirements of an information object, specifically for printed documents or when information is displayed on screen. For this purpose traditional sensitivity markings will still be used, however the sensitivity marking will be derived from the content label (based on protection and release policies in effect at the time of derivation).

A. User and terminal attributes

When an access request for an information object is made, in CPR both the requesting user and the terminal from which the access request is made are considered in the access control decision. The protection requirements and release conditions are expressed in terms of terminal and user attributes. Both user and terminal must meet the requirements expressed by those attributes in order for the information object to be released.

The terminal and user attributes can be chosen with different levels of detail. The CPR model does not prescribe the use of a fixed set of terminal and user attributes, however the terminal and user attributes must be chosen in such a way that the protection requirements and release conditions can be expressed with sufficient detail to allow a useful access control decision to be made. Examples of terminal attributes are: management authority, location, type of communication link between the location of the information object and terminal, available protection mechanisms at the terminal, and indicators of the level of assurance of the available protection mechanisms such as the presence of a trusted platform module (TPM) [4]. User attributes may include: user identity, nationality, organization the user represents, role and clearance level.

B. Content labels and binding mechanism

Given that CPR will be applied in an NNEC environment, it is recommended that the mechanism used for binding content labels to information objects allow for fine-grained ('granular') access control. An example of a binding mechanism that can be used for granular access control is described in [5]. The community of interest governing the creation and management of the information object should define the format of the content label.

IV. General concept

In order to provide a suitable information sharing capability for future NATO missions, the NCI Agency has been working on the design and development of a service-oriented architecture for controlled information sharing in NATO operations; see Fig. 1. The component responsible for handling access requests for information objects (including derivation and enforcement of the access control decision) is referred to as the CPR Enforcement Separation Service (CPRESS). The functionality and composition of the CPRESS is introduced in more detail in section V. The CPRESS is supported by several other services that together enforce the separation of different security domains. Two particularly important services are the Content Inspection Policy Enforcement (CIPE) service, described in section VI, and the NATO Metadata Binding Service (NMBS), described in section VII.

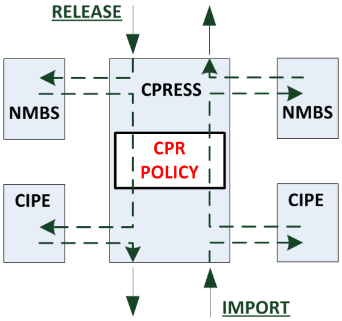

Fig. 1. The CPRESS enforces the CPR policies upon import and release of information objects. The CPRESS interacts with the NMBS and CIPE service.

V. CPR Enforcement Separation Service

A typical CPRESS use case involves the processing of access requests for an information object located at a content server that is protected by the CPRESS. The access request is made by a user from a terminal. CPRESS derives the protection requirements and release conditions by correlating the content label with the protection and release policy, as depicted in Fig. 2. Note that, although it is not shown in Fig. 2, in order to enforce the use of the proper strength of the authentication mechanisms used to authenticate user and terminal, CPRESS must also derive the requirements for the strength of authentication mechanisms from the protection and release policies.

The CPRESS places no restrictions in advance on the administrative domain membership of the users, terminals and content servers involved. However, scenarios that involve multiple different administrative domains require the integration of CPRESS with both a federated capability for identity management and terminal authentication, and a federated capability for the management of attributes (for both users and terminals) and content properties. The design and implementation of the federated capabilities is not within the scope of this paper.

A. Release decision process

When an access request is made the CPRESS authenticates both user and terminal in order to determine the user and terminal attributes. The level of strength and assurance of the authentication mechanism that is used to authenticate the terminal must be commensurate with the level of strength and assurance of the protection mechanisms that will be associated with the terminal based on a positive authentication result. For example, a terminal that is authenticated based on only an Internet Protocol (IP) source address should not be trusted to claim that it offers an encryption capability deemed sufficient for the protection of highly sensitive information. A similar observation applies to the authentication mechanisms that are used for user authentication. CPRESS must therefore also enforce the use of the proper authentication mechanisms. An overview of authentication mechanisms – categorized based on assurance level – that can be used in CPR can be found in [6].

After positive authentication of the user and his terminal, the CPRESS correlates the content properties of the requested resource with the CPR policies, thus resulting in protection requirement and release conditions. Then it matches the protection requirements with the terminal attributes in order to decide whether or not the terminal is able to protect the information (represented by the information object). The user attributes are matched with the release conditions in order to decide whether or not the user is authorized to access the information. Only if both decisions are positive is the information object released to the user at the terminal. This is illustrated in Fig. 2. Note that in some cases it will be possible for the CPRESS to filter information based on the user and terminal attributes so that a 'releasable information object (RIO)' – composed of the releasable parts of the original information object – can be released. Composing the RIO is the core capability of the CPRESS.

Although Fig. 2 does not show the step in which the CPRESS verifies that the proper authentication mechanisms have been used to authenticate user and terminal, this step is necessary for ensuring the overall correctness of the solution. There are different approaches to how to include the requirements for authentication mechanisms in the CPR policies; the requirements can be directly related to content properties (or indirectly related as part of predefined assurance levels that are associated with content properties) or to the user and terminal attributes that the authentication mechanisms aim to verify. Enforcement of the requirements for the authentication mechanisms used by user and terminal can be done explicitly by refusing authentication if the mechanism prescribed by the CPR policies is not supported by user or terminal, or implicitly by treating the authentication mechanism used as an additional user or terminal attribute that is matched against release conditions and protection requirements; if it does not match, the attributes that it aims to verify are discarded.

B. Importing information objects

The opposite of the release decision process is the import of information objects into the domain that is protected by the CPRESS. The import of information can be initiated from the domain protected by the CPRESS, or by an external user who

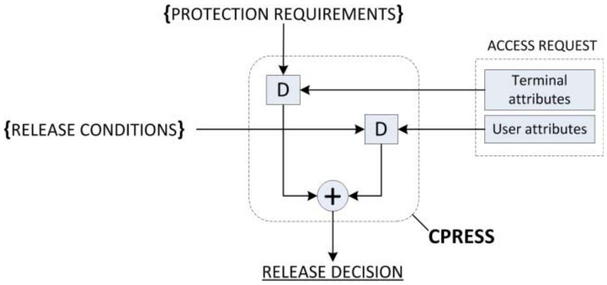

Fig. 2. The CPRESS makes two access control decisions (D) in order to arrive at a release decision; the release decision is positive only if the terminal is able to protect the information AND the user is authorized to access the information.

wants to upload information or initiate a push of information to another user. The enforcement of the CPR policies by the CPRESS will pertain to determining how to protect the imported information, whether or not a user is allowed to create information with a certain set of content properties and upload it to the internal domain, and in the case of a push of information to another user, whether or not the latter user is allowed to receive the information object.

When the user who initiates the import is not a member of the same administrative domain as the CPRESS, the CPRESS depends on a federated capability in order to determine the authorization of the user (and terminal) and the protection requirements for the imported information object that may have foreign content properties. The protection requirements can be derived locally after first mapping any foreign content properties to local content properties. However, this requires that such a mapping exist, and the local protection policy must be such that a level of protection can be derived that is equivalent to the level achieved in the domain of the federated partner. If these prerequisites are met the local equivalent content properties can be stored in an 'alternative' content label (see [5] for an approach). If a mapping between content properties does not exist, the federated capability must allow for a lookup of the protection requirements with the appropriate federated partner after which a translation to local equivalent protection requirements may be required.

C. Data-flow model

The general design of the CPRESS is based on the Extensible Access Control Markup Language (XACML) version 3.0 data-flow model [7].

The main components of the XACML version 3.0 data-flow model are: *Policy Decision Point (PDP)*, the system entity that evaluates applicable policy and renders an authorization decision; *Policy Enforcement Point (PEP)*, the system entity that performs access control by making decision requests and enforcing authorization decisions; *Policy Administration Point (PAP)*, the system entity that creates a policy or policy set; and *Policy Information Point (PIP)*, the system entity that acts as a source of attribute values.

Fig. 3 illustrates the interaction between the components that takes place when an 'access requester' (i.e. the 'subject') issues an access request for a specific resource. The XACML data-flow model does not place any restrictions on the communication protocols that can be used for the interaction between the components, e.g. communication between the PDP and PIP/PAP may be facilitated by a repository.

In a typical use scenario for CPRESS, a content server provides resources labelled with content properties. The CPRESS mediates access requests for such resources. CPRESS delegates authentication of the user to the identity provider. User attributes are stored and managed at the identity provider. The identity provider issues security tokens that include user attributes; the security tokens are verified by the CPRESS. Users belonging to a different administrative domain than the CPRESS must authenticate through a federated identity management capability.

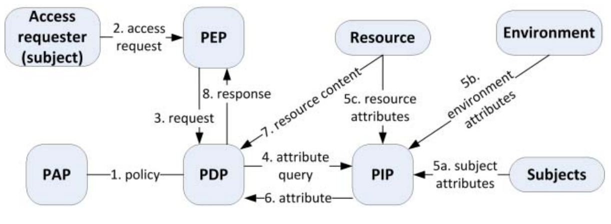

Fig. 3. Simplified illustration of the interaction between the components of the XACML data-flow model

Terminal attributes are stored and managed at the terminal capabilities server. The CPRESS authenticates the terminal from which the access request is made and retrieves terminal attributes (including capabilities but also other attributes such as 'management authority') from the terminal capabilities server. Here it is assumed that a federated terminal authentication capability exists.

Fig. 4 shows how the CPR components fit into the XACML version 3.0 data-flow model.

The access request for a resource located at the content server is made by a user at a given terminal. This means that the user takes on the role of the 'access requester' as identified in Fig. 3. The CPRESS makes an access control decision with respect to the requested resource for both the user and the terminal. Therefore, although the access request is made by the user, for the purpose of the access control decision the user and terminal can both be considered to be subjects that request access to the same resource. This means that the 'subject attributes' as identified in Fig. 3 correspond to both user and terminal attributes.

The CPRESS consists of both a PDP and a PEP. The PDP is the component responsible for evaluating content properties and attributes against the CPR policies and deriving an authorization decision. The PEP is the component that enforces the access control decision. Although in the figure the PDP and PEP are depicted as independent systems, these components can be combined on one system. The access request made by a user from a given terminal is mediated by the PEP component of the CPRESS. The PEP component authenticates the terminal and initiates user authentication, which is further delegated to the identity provider. After user and terminal authentications have been completed, the PEP retrieves the resource. The content properties are extracted by the PEP from the resource

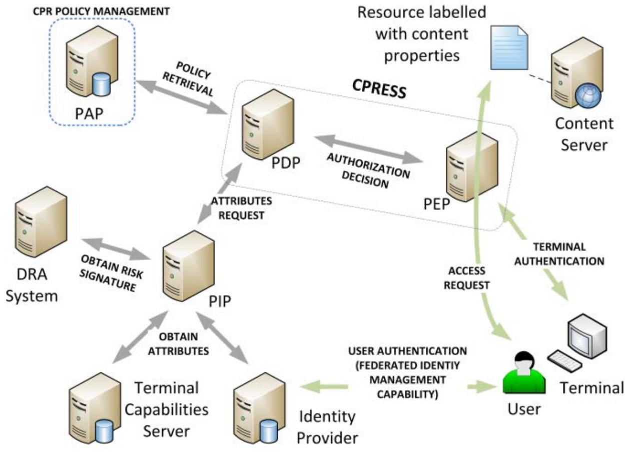

Fig. 4. Illustration of how the main components of the CPR model fit into the XACML data-flow model

attributes with which the resource is labelled and are sent to the PDP with the request for access (note that it is also possible to introduce a separate resource mediator component that retrieves the resource and extracts the content properties).

The CPRESS interfaces with a PAP that stores the CPR protection and release policy. The PAP can be a separately managed system that interfaces with more than one CPRESS. The management of the CPR policies takes place at the PAP. The CPRESS requests user and terminal attributes from a PIP. The PIP interfaces with the terminal capabilities server and the identity provider, from which it obtains the required attributes.

The CPRESS can also request specific environment attributes such as the risk signature of the network, which can be provided by a Dynamic Risk Assessment system [8], which interfaces with the PIP. This enables the CPRESS to provide risk-based adaptive access control [9].

1) Handling of subject attributes

Note that in a federation of systems an access request may involve users and terminals from a different administrative domain than the domain in which the CPRESS operates. In such cases there are two options for handling subject attributes: 1) User and terminal attributes are presented during authentication (e.g. in case of user attributes they can be included in the security token issued to the user by the identity provider from the user's administrative domain); in this case the attributes can be included by the PEP in the authorization request. 2) The local PIP obtains user and terminal attributes from the remote PIP (which is located in the user's domain and interfaces to the identity provider in the user's domain), based on user and terminal identity.

The choice of which option to implement in the federated case depends on how user and terminal authentication is implemented locally, the availability of an interface (and trusted channel) to the remote PIP, and scalability.

2) Handling of the requested resource

In the CPRESS the resource can in principle be handled only by the PEP, meaning that step 7 in Fig. 3 does not take place. This is because in order to make an authorization decision it is sufficient for the PDP to have access to user and terminal attributes, and to the content properties of the information object for which an authorization decision has to be made. In such cases, verification of the integrity of the requested resource and the resource's label is done by the PEP. Although it is possible to have the verification (including the extraction of content properties) done by the PDP – which would then have to retrieve the resource – this design choice may not be scalable with respect to the PDP's computing resources if the PDP has to support multiple PEPs. Also, it must then be guaranteed that the instance of the resource that is verified by the PDP is identical to the instance handled by the PEP at the moment that the PEP enforces the access control decision. Such a guarantee can be achieved by having the PEP 1) facilitate the PDP retrieving the exact instance of the resource that the PEP is handling so that the PDP can verify it; or 2) verify the integrity of the instance of the resource that the PEP is handling before enforcing the access control decision.

The first option is not only a clear deviation from the XACML data-flow model, but also requires interaction of the PDP with the PEP's memory resources, which may in itself lead to scalability issues in case the PEP has to handle multiple requests for access. The second option defeats the purpose of having the PDP verify the integrity of the resource. For these reasons and the argument of scalability with respect to the PDP, the PEP is responsible for verification of the integrity of the resource (and the label).

VI. Content Inspection Policy Enforcement

In addition to enforcement of the CPR policies, the information objects should be checked for malicious or otherwise unwanted active content. The CIPE service implements the required content inspection through the use of content filters. The CIPE framework (CIPEF) [10] provides an interface to the content filters and ensures that the content filters are executed correctly as information objects are managed and scheduled through the CIPEF; see Fig. 5. The role of a content filter encompasses the following:

- Identification of an information object (i.e. to determine the type of content);
- Verification of an information object (i.e. to verify the content type after identification and to determine if the information object contains elements that are not permitted for information exchange);
- Transformation of an information object (i.e. modification of data to remove or transform data elements that are not permitted for information exchange).

The types of content filter and the rules that they enforce are determined by the specific information exchange requirements and the information management policy (which also covers the information exchange (security) policy).

A. Role of CIPE proxies

A CIPE proxy interface is the boundary between the CIPE service and the information exchange architecture and can handle protocol and content mediation between the data source and the CIPEF. The CIPE proxy can be implemented in one of several ways, including input and output file stores, inter-process communication, and transport protocol proxy, e.g. an HTTP proxy.

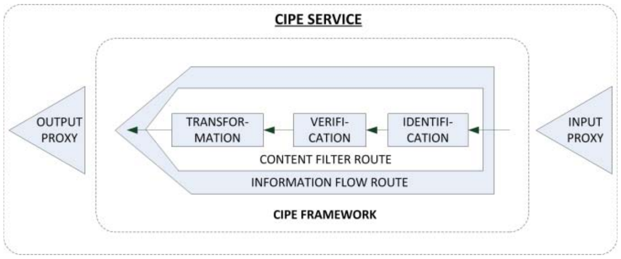

Fig. 5. Content Inspection Policy Enforcement service

The choice for a type of proxy is dependent on the system architecture. If the CIPE service is implemented in a guard device, the CIPE proxies may coincide with the transport protocol interface offered by the guard if the guard is composed solely of the CIPE service. If the CIPE service is one of more services running on the guard, then the CIPE proxies are implemented separately. Here one could choose to implement the same protocol proxies as the guard offers, in order to be able to later move the CIPE service to a separate physical device without further modification of the architecture.

Two CIPE proxies are distinguished: the *input* proxy and the *output* proxy. The design of the CIPEF is such that each input and output proxy corresponds to a separate inbound and outbound channel respectively. Multiple CIPE proxies can exist and a CIPE filtering policy is defined for each inbound and outbound channel pair. This allows for separation of different information flows based on protocol, file store (e.g. USB media import), and even physical interfaces.

B. CIPEF in the CPRESS

In order to separate the processing of internal objects (objects residing in the internal domain that is protected by the CPRESS) from external objects (objects residing in an external domain that enter the internal domain), it is proposed to have at least two CIPE services: one service inspects external information objects that enter the internal domain, and the other service processes internal information objects that are candidates for release to the external domain.

If the CPRESS protects more than one internal domain from potentially more than one external domain, the number of CIPE services should be adjusted accordingly.

Interaction between CPRESS and CIPE can take place before or after the enforcement of the CPR policies. In order to reduce the surface of an attack against the CPRESS when importing information objects, CIPE is executed before further processing of the information object by the CPRESS. However, in the case of release of information objects, given that the information objects reside in the internal domain, it is preferable to protect the availability of the information exchange capability and therefore limit as much as possible the processing of potentially not-releasable data by the CIPE service. Therefore in the case of release of information objects, the CPR policies are enforced first.

VII. NATO METADATA BINDING SERVICE

In order for the CPRESS to properly enforce access control, an information object must be labelled with content properties in a trustworthy way that preserves the integrity of the information, its properties and the binding.

The NATO Metadata Binding Service (NMBS), as shown in Fig. 6, provides a flexible and extensible service that can be used to support the binding of metadata to information objects, using a binding mechanism of a particular strength. The NMBS supports service clients by providing metadata enumerations (of which ‘content property catalogues’ are an example), binding metadata to information objects (in order to produce binding data) and validating binding data. When binding metadata to information objects the NMBS will handle the metadata in the form of a label; the NMBS can therefore also be regarded as a ‘labelling service’.

The NMBS can be used from desktop clients and central servers within a domain, but also at domain boundaries where the NMBS can support the mapping between different metadata policies that apply to the domains involved. The NMBS makes use of existing Core Enterprise Services such as Security Services (including public-key infrastructure (PKI) services) and the Enterprise Directory Service.

The design of the NMBS currently does not include interaction with a repository for the storage of information objects and binding data. However, in certain use cases it may be more appropriate for the NMBS to return an information object or binding data by reference and store the actual information objects in a repository, e.g. if the requestor or target of an NMBS operation is not authorized to handle a specific information object or binding data. In addition, a service client may wish to explicitly store binding data in a repository. Possible benefits of upgrading the NMBS to interact with a repository service will be assessed in future investigations.

The NMBS is located in a secure domain and CPRESS invokes the NMBS *Verify* operation after information object retrieval (on behalf of an authenticated user and terminal). The NMBS verifies that the label is correct, is of allowed format (and semantics/values) and verifies the signature. If re-signing is required after an information object is transformed into a RIO, then the NMBS *Set* operation can be invoked. Note that Fig. 6 shows a connection between the NMBS and CIPE. The NMBS calls the CIPE service only when a user creates an information object and invokes the *Set* operation; before the NMBS signs an information object, it must first be verified that its contents do not violate the information management policy. The evaluation of the binding in the sense of determining which content properties are bound to which information objects is executed by the CPRESS.

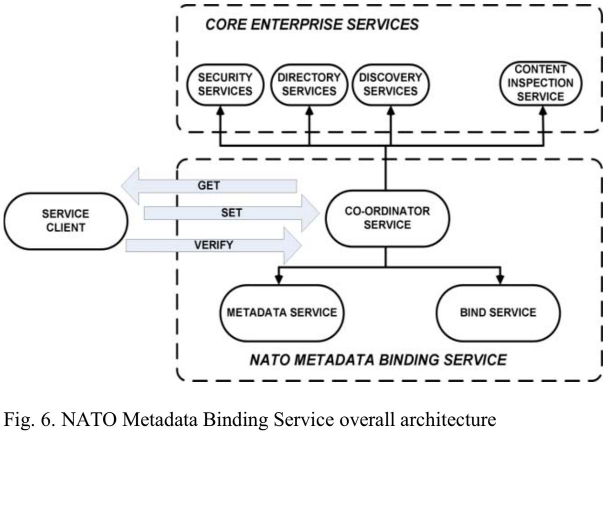

Fig. 6. NATO Metadata Binding Service overall architecture

VIII. Related work

In parallel with the development of the CPR architecture at the NCI Agency, other architectures based on ABAC and the XACML data-flow model are being developed by NATO nations. One such example is the Secure Access Management for Secret Operational Networks (SAMSON) architecture [11], developed by Defence Research and Development (Canada). Although SAMSON and the CPR architecture have similar functional components, CPR extends the SAMSON architecture by using terminal attributes in addition to user attributes, and content labels (containing content properties) instead of security labels. The addition of terminal attributes and content properties in CPR introduces additional challenges in a federation of systems and also leads to the requirement to develop a verifiable (i.e. formal) and robust security policy model that will form the basis for the CPR policies and the enforcement of access control in CPR, as proposed in [12]. The use of encryption is not discussed in this paper; initial results of our research on the use of attribute-based encryption (ABE) in CPR are presented in [13]. This led to the CPR architecture being extended to include the possibility to enable end-to-end encryption of information objects based on ABE. The advantage of this extension is that the CPR policies can be enforced through cryptographic access control, even in situations in which mediation of information objects by the CPRESS cannot be guaranteed.

IX. Conclusions

A number of conclusions and recommendations were identified during the course of work on the CPRESS design and the related experimentation. First, each of the different communities of interest in NATO must be encouraged to standardize a set of content properties for its community in coordination with its stakeholders. In order to facilitate the process of determining appropriate content properties, the use of machine learning techniques (that can be applied to existing content) should be considered. Wide deployment of the CPR model also requires the provision of all standardized sets of content properties in 'content property catalogues' that are accessible by information creators. The development of content property catalogues needs to be aligned with other activities performed within the NATO information and knowledge management community, such as the development of the NATO Metadata Discovery Service. How such content property catalogues will be made available as a service to user applications in the network is still an open question, although as noted in this paper a potential candidate to offer such a service is the NATO Metadata Binding Service (via the *Get* operation; see Fig. 6). A related challenge is the establishment of a comprehensive set of user and terminal attributes, which could support the definition of all required security policies.

The operation of CPRESS in a federated environment brings specific challenges such as the realization of a federated terminal authentication capability in support of CPR, and the management of attributes and content properties in a federation. These aspects require further research.

There are several options for how to implement the CPRESS, NMBS and CIPE service. For example, some of the operations executed by CPRESS and NMBS can be considered to be filters and as such they could be implemented by the CIPE service. This however will require close coordination between all services with respect to attributes and content properties. Further research is required to determine the impact of the coordination as well as the potential benefits of such an architecture.

The experimentation phase revealed that a functional CPR architecture can, to a great extent, be implemented based on currently available commercial and open-source technology. The challenge however in realizing an operational deployment of the CPR architecture in NATO is threefold. First, there is a requirement to develop and manage standardized content property catalogues. Second, the CPR concept relies on the availability of trusted systems and software that can enforce mediation of all information exchange by a CPRESS and proper separation of local environments on terminals. An implementation of CPR therefore depends on the advancement of such technology. Lastly, CPR and the use of content labels in particular is not covered by existing NATO policy, which may hamper the accreditation process of an operational implementation. The NCI Agency has started to identify the NATO policy documentation that is affected by the concept of CPR in order to develop recommendations regarding NATO policy updates, however in order to advance the adoption of CPR a feasibility study must be undertaken that addresses all of the challenges above.

Finally, CPR depends on the ability to assign content properties to information objects upon creation of the object. We therefore propose investigating the feasibility of developing a labelling-assistance application (LAA), which not only allows a creator of an information object to create a label, but also assists the creator in selecting the appropriate content properties from a content property catalogue. When use of traditional sensitivity markings is required, e.g. for printed documents, the LAA could also support users in determining the appropriate sensitivity marking based on the selected content properties.

References

[1] K. Wrona and G. Hallingstad, "Controlled information sharing in NATO operations," in *IEEE Military Communications Conference (MILCOM)*, 2011, pp. 1285–1290.

[2] S. D. C. Di Vimercati, S. Foresti, and P. Samarati, "Access control policies and languages," *International Journal of Computational Science and Engineering*, vol. 3, no. 2, pp. 94–102, 2007.

[3] A. Domingo and H. Wietgrefe, "A NNEC-compliant approach for a Future Mission Network," in *Proc. of the Military Communications Conference (MILCOM)*, 2012.

[4] Trusted Computing Group (TCG), "Trusted Platform Module Library Part 1: Architecture," 2012.

[5] S. Oudkerk, I. Bryant, A. Eggen, and R. Haakseth, "A Proposal for an XML Confidentiality Label Syntax and Binding of Metadata to Data Objects," in *NATO RTO Symposium on Information Assurance and Cyber Defence*, 2010.

[6] W. Burr, D. Dodson, E. Newton, R. Perlner, W. Polk, S. Gupta, and E. Nabbus, "NIST Special Publication 800-63-1 Electronic Authentication Guideline," Gaithersburg, MD, 2011.

[7] OASIS, "eXtensible Access Control Markup Language (XACML) Version 3.0," 2010.

[8] K. Wrona and G. Hallingstad, “Real-time automated risk assessment in protected core networking,” *Telecommunication Systems*, vol. 45, no. 2–3, pp. 205–214, Jan. 2010.

[9] S. Kandala, R. Sandhu, and V. Bhamidipati, “An Attribute Based Framework for Risk-Adaptive Access Control Models,” in *Proc. of the 6th International Conference on Availability, Reliability and Security*, 2011, pp. 236–241.

[10] K. Wrona and G. Hallingstad, “Development of High Assurance Guards for NATO,” in *Military Communications and Information Systems Conference*, 2012.

[11] D. Charlebois, “Secure Access Management for SECRET Operational Networks (SAMSON),” Ottawa, Canada, 2009.

[12] A. Armando, S. Oudkerk, S. Ranise, and K. Wrona, “Content-based Protection and Release for Access Control in NATO Operations,” in *Proc. of the 6th International Symposium on Foundations & Practice of Security (FPS)*, 2013.

[13] S. Oudkerk and K. Wrona, “Cryptographic Access Control in support of Object Level Protection,” in *Proc. of the Military Communications and Information Systems Conference (MCC)*, 2013.