Protected and Controlled Communication Between Military and Civilian Networks

Anders Fongen
Norwegian Defence Research Establishment
Norway
anders.fongen@ffi.no

***Abstract*—The controlled and protected communication between civilian and military computer nodes is the objective of this paper. The release of unclassified military information to Non-Governmental Organizations (NGOs) may improve the safety and effectiveness of their operations. The information exchange must meet several requirements though, related to military tactics, the impartial status of the NGO and international *jus in bello*. The paper proposes a framework that both protects communication and controls the access to information resources. A prototype based on the framework has been built and was evaluated during the CoNSIS experiment in June 2012.**

Keywords-CiMi, Identity management, Authentication

I. Introduction

The presence of non-governmental organizations (NGOs) in a war zone is frequently seen, and their operations may be safer and more efficient through communication with military forces. Military information about safe routes, road conditions and observations regarding the situation for the population may be sent to the NGOs. Positions and movements of NGO vehicles and personnel may be sent to the military forces in order to avoid inadvertent attacks.[1]

The information exchange must not blur the impartial status of the NGOs and must not weaken the protection of NGOs by international laws of war. NGO equipment must never convey or relay military information, and never provide information of value for the military operation. The NGO should not possess military hardware or participate in proprietary military communication protocols.

From these perspectives, the detailed control of the information exchange becomes an essential property. In this paper, the proposed technical elements of interconnection, protection and control will be described and discussed.

The contribution of this paper is a separation and control framework for the "minimal" interconnection of networks, where only selected and essential services are allowed to cross the CiMi interface. The framework relies to a large extent on the Identity Management system previously presented in [2], but leverages that system into an enterprise context where additional technologies like IPSec and the XMPP protocol complements the Identity Management and offers a more "hardened" system. Besides, the discussion of requirements set on behalf of impartial and civilian NGOs has not been observed previously in the context of computing security research.

The remainder of the paper is organized as follows: The next section articulates the technical non-functional requirements of the interconnection, followed by Section III where the proposed system configuration is described. Section IV gives a general introduction of Identity Management services, which are central to the proposed solution framework. Sections V-VII present the IdM prototype used for the evaluation experiment. Section VIII gives a brief presentation of the mechanisms bridging the IdM service invocation environment with the classified SOA environment. Section IX presents a set of problems related to the unconventional use of COTS products. Section X presents the experimental environment in which the framework was evaluated, and the paper finishes with a section containing some concluding remarks.

II. Technical requirements

The functional requirements for a Civilian-Military (CiMi) communication arrangement may be expressed in the following manner:

A. COTS equipment and protocols

The NGO should avoid the use of military communication equipment from reasons of impartiality and cost. A laptop computer or a smartphone is able to communicate over a WiFi link or a cellphone connection. Where possible, public communication service should be used, even though the military end of the connection would also need to link to a similar service. For longer ranges in environments without a communication service, civilian radio equipment with computer interface may be used.

B. Protection of communication channel

The CiMi connection must be a black network, i.e. it can run through any unprotected link. This supports the utilization of public network services or private radio links without any link crypto requirements.

The end-to-end connection (possibly spanning several different links) must be protected with cryptographic equipment/software which is available for non-military use, i.e. an IPSec tunnel protected with AES.

C. Robustness of separation (fail-close)

The separation of the NGO and the military equipment should have the *fail-close* property (also called *fail-safe*). Fail-close means that in the event of a failure, system security should be preferred before connectivity. Any filtering or control mechanism should operate in a *deny-allow* order where the default action is to deny service.

D. Authentication of participants

Participants in the communication should be fully identified before or during the service. Authentication is the basis for resource control and auditing, and normally requires a registry of users and services where their identity is associated with the necessary credentials. Authentication across the CiMi interface should not require that the identities are registered on both sides: A *Cross Domain* mechanism should be in place where a trust relation between the registration authorities should allow mutual authentication across the interface without the need for multiple registration of identities.

E. Role-based access control

Since authentication does not require local registration of an identity, (cf. previous section) the decision to allow or deny participation in the service transaction cannot rely on the identity, but rather on roles or attributes associated with the identity. Role Based Access Control [3] should be the basis for the access control decisions, which enables the owner of a service to reserve its use for clients which possess certain roles. RBAC preserves the autonomy of domains and let them define and enforce their own independent security policies.

F. Confidentiality labeling

In the classification hierarchy found in military information management there is a need to decide if information kept in classified systems can be released for use on lower classification levels and even released to an NGO. One approach to achieve this is by means of confidentiality labels. They are cryptographically bound to the information object and can be automatically inspected by a *guard*. The guard is situated between networks of different classification levels and transfers object from high to low classification based on the confidentiality label and a transfer policy. The guard provides an isolation between two military networks and adds to the separation between the unclassified military network and the NGO.

III. The prototype configuration

For an experimental evaluation of these principles (cf. Section X) a prototype was developed with the following services in mind:

A. Protected service invocation

A client in the NGO network should be able to invoke a positioning service in the classified network, and to receive the GPS coordinates of a mobile military unit. The service requires mutual cross domain authentication across the CiMi interface, role based access control decisions, and data inspection by the guard in order for the invocation to succeed.

B. Secure chat

The mobile client may write text messages to other users on a chat client program. The chat messages must be protected in the same manner as service invocation messages using the same cryptographic mechanisms. There is no need for end-to-end authentication, and a simple authentication mechanism provided by the chat server is sufficient. The chat message service covers users connected to the NGO network or the unclassified military network, but the chat server will reside in the military network.

C. Configuration details

Figure 1 outlines the structure of the prototype. It consists of the following actors:

- An Android smartphone, acting as an NGO terminal for chat and protected service invocation.
- A chat server for the XMPP chat protocol. This server will forward both chat messages and service invocation messages.
- Two Identity Providers (IdP), one for the NGO domain and one for the military domain. They provide identity information for authentication operations. The details of the IdP will be explained in Section IV.
- An application server, residing in the military domain, hosts application services or proxies for Web Services.
- A SOAP guard, which connects the military classified and unclassified networks. It ensures that only correctly labeled data is passed from the classified to the unclassified part.
- Other chat clients which use the XMPP protocol. They are connected to the XMPP server.

The XMPP protocol is used for the transport of chat messages as well as messages for the protected service invocation. Clients or services need to connect (and log in to) the server in order to participate in any of the two services. The XMPP chat server is the only connection between the NGO nodes and the military nodes, and there is no IP route between the two networks. The figure also shows that connections outside the physical control of a wired military network is protected with IPSec tunnels.

IV. Introduction to Identity Management

(Most of the text in the following 4 sections are previously published in [2]).

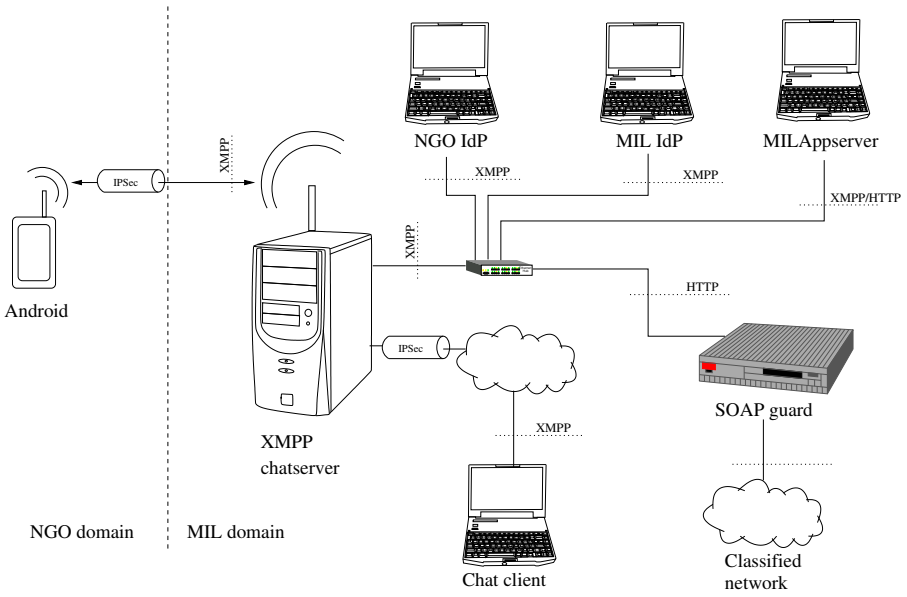

Figure 1. Outline of the experimental prototype for the demonstration of CiMi communication

Identity Management (IdM) are collection of services and procedures for maintaining subject information (key pair, roles) and to issue credentials for the purpose of authentication, message protection and access control. From the client perspective, the credentials issued by the IdM services enables it to access many services inside a community under the protection of mutual authentication and encryption. From the server perspective, IdM enables it to offer credentials to clients in order to provide mutual authentication.

The arrangement of an IdM resembles the Public Key Infrastructure (PKI), in the sense that a Certificate Authority (CA) can issue *public key certificates* which binds an identity to a public key in a way that can be validated by a *relying party*. The binding is made by the CA's signature using a well known and trusted key. The role of the CA, called *trusted third party*, is widely used when making arrangements between parties that have never met before.

The traditional organization of a PKI is to issue public key certificates with a long lifetime, typically 1 year. In the event that the key need to be invalidated before expiration, it need to be *revoked*. Revocation information needs to be disseminated to all relying parties in the form of *revocation lists* or *online status providers*. There are two main reasons why a traditional PKI is not a viable solution for identity management: First, the distribution of revocation information is costly in terms of bandwidth and connectivity requirements, and secondly because the public key certificate does not contain information about the subject necessary to make access control decisions.

The requirements of an IdM (distinct to the requirements of a PKI) should be:

- The IdM should issue short term credentials so that distribution of revocation info becomes unnecessary.
- The IdM should include role/attribute information about a subject to support access control decisions etc.

The decision to avoid distribution of revocation information is based on a comprehensive study of scalability properties in commercial PKI implementations [4]. The conclusion of that study is that short lived credentials generate less network traffic, have less connectivity demands, scales better and make the validation operation more intuitive.

A. Federated Identity Management

Several federated IdM schemes have been developed, some of which offer single sign on (SSO) for web clients [5], [6], [7]. The SSO protocols exploits the redirection mechanism of HTTP in combination with cookies and POST-data so that an Identity Provider (IdP) can authenticate the client once and then repeatedly issue credentials for services within the federation. This arrangement requires IdP invocation for each "login" operation, and does not offer mutual authentication, i.e., no service authentication.

In the situation where the client is an application program (rather than a web browser), there are more opportunities for the client to take actively part in the protocol operations, e.g., by checking service credentials, contacting the IdP for the retrieval of own credentials, caching those credentials etc. The research efforts presented in this paper assume that the clients enjoy the freedom of custom programming.

The usual meaning of the word "federated" is that several servers share their trust in a common IdP for subject management and authentication. It does not necessarily imply any trust relationship between independent IdPs so that they can authenticate each others' clients. For the following

discussion, we will call the group of clients and services which put their trust in the same IdP as a *community of interest* (COI). A trust relation between independent IdPs is called a *cross-COI relation*.

B. Mobile and Federated IdM requirements

An essential property of an IdM is its ability to integrate with other components for management of personnel and equipment.

- An IdM should be able to use resources from the existing PKI (keys, certificates, revocation info) and offer its services to different platforms, with different presentation syntax and for different use cases.
- An IdM should also be able to tie trust relations with other IdMs in order to provide accommodation for guests and roaming clients.
- An IdM should support protocol operations for mutual authentication.

For IdM used in mobile systems, there are requirements related to the resource constraints found in these systems:

- A IdM for mobile operations must use the minimum number of protocol operations, use small PDU sizes and must allow the use of caches.

C. The relation between IdM and Access Control

Services can enforce access control on the basis of the *identity* of an authenticated client, or based on *roles* or *attributes* associated with the client. For the purpose of the accommodation of roaming users, it is absolutely necessary to make access control decisions based on roles/attributes, not identity. Identity based access control requires that all roaming clients are registered into the guest IdM, which is an unscalable solution.

The principles of *Role/Attribute Based Access Control* (RBAC/ABAC) are well investigated [3]. The names and meaning of the roles/attributes that are used to make access decisions must be coordinated as a part of an IdM trust relationship. For that reasons, the number of roles/attributes used for access control needs to be kept low.

It is the obvious responsibility of an IdM to manage the roles/attributes of a subject, some of which may enter into access control decisions, others be used by the service to adapt the user interface etc. The presence of subject attributes is the main functional difference between IdM credentials and X.509 public key certificates.

V. The GISMO IdM architecture

For the purpose of authenticated service provisioning in military tactical networks (meaning wireless, mobile, multi-hop, multicarrier networks), an Identity Management system has been developed under the project name "GISMO" (General Information Security for Mobile Operation). The system has been previously presented in [8], [9], so its properties are only briefly listed here:

Subject Distinguished Name
Subject Public Key
Subject Attributes
Valid from−to
Issuer Distinguished Name
Issuer Public Key
Issuer's Signature

Figure 3. The structure of the Identity Statement

- It uses short lived *Identity Statements* containing the subject's public key and subject attributes. No revocation scheme is necessary. Identity Statements are issued by an Identity Provider (IdP).
- Cross COI relations are represented by ordinary identity statements issued from one IdP to another.
- IdPs can issue *Guest Identity Statements* when presented with an Identity Statement issued by an IdP with with which it has a Cross COI relation. A guest identity statement contains the same information, but is signed by a different IdP.
- Authentication takes place either through a signature in the service request, or through the encryption of the service response.
- It supports Role/Attribute Based Access Control (RBAC/ABAC) through the subject attributes.
- Employs, but encapsulates an existing PKI. Clients never see X.509 certificates or revocation info.
- Identity Statements are cached and re-used during its lifetime. An IdP is invoked to issue Identity Statements, not to verify authenticity.
- There is loose coupling between IdP and services/clients, and between COIs. Very little redundant registration is necessary.

Figure 2 illustrates the concepts and components of the GISMO IdM. Identity establishment, key generation and key certification happens in the (existing) PKI. Related to a CA (Certificate Authority) domain there are several Communities of Interest (COI) with one IdP common to all members of that community.

The IdP issues signed *Identity Statements*. The structure of the Identity Statement is shown in Figure 3.

Members of a COI only trust the signature of their IdP, so an Identity Statement (signed by the IdP) is not valid outside the COI unless there exists a *cross-COI Identity Statement* which links the signature of the foreign IdP to the trusted IdP. More on that later.

A. Cross COI relationships

Any client will likely be a member of several COIs, reflecting the diverse tasks and responsibilities of a worker or a soldier. It is not convenient to manage the client's

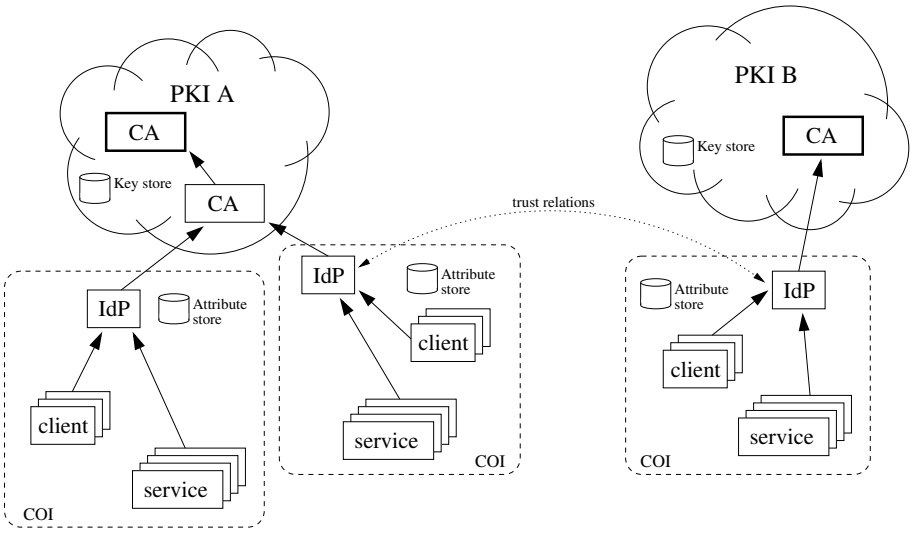

Figure 2. The functional components of a federated IdM. Observe that the IdP serves one single COI, and the trust relations are formed between COIs, not domains. Key management is handled by the PKI whereas the attribute management is done by the IdPs on the COI level

TABLE I
ABBREVIATIONS USED IN THE FIGURES

Client X_a	Client X of COI a
IdP_a	Identity provider of COI a
PKI_a	Validation services in domain a
Server F_b	Server F in COI b
$(Id_x)_a$	Identity statement for identity x, issued by IdP_a
$(msg)S_x$	Message msg signed with private key of x
$(msg)E_x$	Message msg encrypted with public key of x

key pairs, attributes etc. in every COI. Most of them will naturally belong to one COI, e.g., their national military unit or the employing department, and could be regarded as "guests" in other COIs.

The ability to authenticate across COI borders is believed to be an essential requirement for a modern IdM. In the GISMO IdM, this problem has been solved by the use of *Guest Identity Statements*. One IdP can issue a Guest Identity Statement if presented for an Identity Statement issued by an IdP with which it has a trust relationship. The trust relationship is represented by a pair of *cross-COI Identity Statements* issued from one IdP to the other.

During invocation of a service in the foreign COI, the client presents the Guest Identity Statement as a part of the authentication process.

Figure 4 shows the interaction between the client and the IdPs during the issuance of identity statements. Please observe that the cross-COI identity statements are issued asynchronously with regard to the client operations, but handed back to the client during issuance of a guest identity statement. Abbreviations used in the figure are explained in Table I.

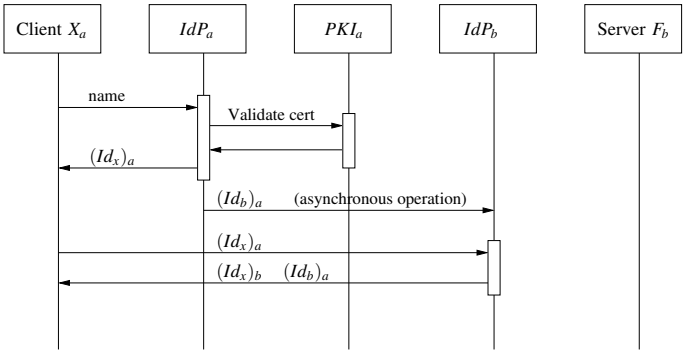

Figure 4. The identity statement issuing protocol. The IdP of COI A, termed IdP_a, issues a "native" identity statement to the client, which is given to IdP_b, which in turn issues a guest identity statement. The term PKI_a denotes a set of certificate validation services in COI a.

VI. SERVICE INVOCATION

IdP operations and service invocations are using serialized Java objects (called *POJO*) as PDUs which opens up interesting opportunities: The client may simply send a parameter object to the server containing the parameter values, and the *class* of the object identifies the service method. This arrangement eliminates the need for a separate scheme for service addressing and also eliminates the need for separate stub/skeleton compilation.

In the server, a single service endpoint hosts all services. This is possible since we do not address the service through a URL, but through association with the parameter class. The service point is a "dispatcher" service, and the serialized parameter object included in the request operation controls the dispatching process. The services are loaded dynamically from a JAR file repository at servlet startup and deployed

through class introspection, no configuration file editing is necessary. Consequently, the deployment of services requires less configuration than e.g. ordinary Java servlets.

A. Authentication dependent on server state space

The authentication mechanisms assure the identities of the client and service during service invocation. Many different authentication protocols can be incorporated into GISMO IdM as long as they employ a public key pair corresponding to the information in the Identity Statement. It is also a requirement that the authentication can be piggybacked on the service request and should not generate separate PDUs. Two protocols have been implemented in GISMO IdM:

1) In those cases where the request must be authenticated *before* the service execution, a replay protection must be in place. Replay protection requires the server to remember past requests (by their Nonce) for a while, so a clock synchronization scheme and a non-volatile stable storage must be in place (since past requests must be be remembered also across server incarnations). These requirements are rather costly.
2) In the case of a *stateless* service, where the execution of a service request does not alter the state of the service, replay protection is not necessary. A request should be signed by the client in order to protect the integrity of the message, but no Nonce for request replay protection is included. The response is *encrypted with the client's public key*, making it useless for everyone but the holder of the private key. To a stateless server, replayed requests are not a threat and protection is not needed. Requests still need a Nonce for reasons of response replay protection, but that does not increase the state space in the server.

Figures 5 and 6 shows the two variants as an interaction diagram. The interactions shown with dotted lines are related to IdP operations and discussed in more detail in Figure 4.

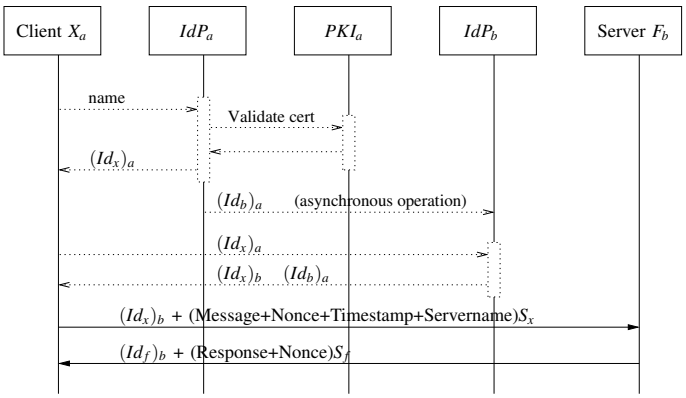

Figure 5. The authentication protocol for the stateful service. Both the request and response are signed with the sender's private key as a part of authentication process. A timestamp, a nonce and the server's name is included for replay protection.

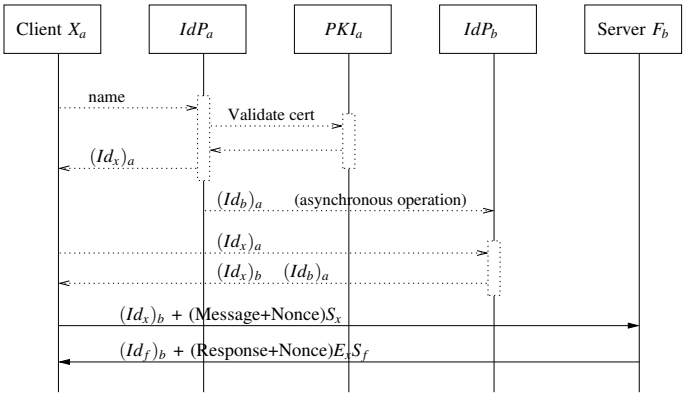

Figure 6. The authentication protocol for the stateless service. Requests are not reply protected since this is not considered as a threat, but the response need to be protected for reasons of response replay and information compromise. For the sake of integrity protection, the request is signed. The encryption of the response is a part of the authentication scheme, not a privacy measure.

B. Authentication during Identity Statement Issuance

For privacy protection, authentication also takes place during Identity Statement issue operations. The client simply signs the request with its private key. If the requested Identity Statement contains the corresponding public key the client is regarded as authenticated. For replay attack protection, the response is encrypted with the public key of the client, which also serve to protect the potential privacy of the subject attributes.

VII. Messaging protocols

In a wired private network where capacity and reliability suffice, and there exist IP routes between the nodes that wish to communicate, the HTTP protocol works just fine for IdP operations and service invocations. For mobile networks this is not necessarily the case: they are slow, unreliable and consists of several partitions connected with application level gateways (from reasons of security and traffic control).

In the context of this experimental study of the GISMO IdM, an XMPP (eXtensible Messaging and Presence Protocol) network was already in place for chat communication. Through the XMPP routers (working as application gateways) otherwise isolated networks (where no IP route exists between them) can exchange chat messages.

A. Service provision by mobile units

A messaging system creates reachable endpoints for nodes which are disconnected at the IP layer. Nodes which reside behind a NAT unit or a firewall are unreachable from the outside world at the IP layer, yet a messaging system can send them messages. Through the XMPP protocol a mobile node can receive service requests just like any other service provider. The prototype system uses a very simple service

container (not a servlet), which is easily portable to a mobile Android based unit.

B. Access to SOAP based web services

The service invocation mechanisms offered by GISMO IdM employs serialized Java objects (POJO) for its protocol data units. On the other hand, there may be existing Web Services based on SOAP messages that clients wish to invoke.

In order to invoke SOAP services, proxies can be built that translates between POJO and SOAP services. This approach has been studied and tested, and represents an attractive approach. A service which takes the parameter values and passes them to a precompiled web services stub (generated by the WSDL compiler). The return value from the stub is passed back to the caller of the POJO service. Example code lines required for this function are shown below:

```
public class MainClass {
  public Serializable service(WeatherRequest wr,
                          Properties props) {
    try {
      Weather w = new Weather();
      String result = w.getWeatherSoap()
         .getWeather(wr.town);
      return result;
    } catch (Exception e) { return e; }
  }
}
```

This option is also attractive since it gives the developer control over service aggregation and orchestration. One service call to a POJO service need not be passed on as one single web service invocation. Many individual calls may be made, and they may be sequenced or tested in any manner. Aggregated operations are useful because they potentially reduce the network traffic to and from the mobile unit, which is likely to be connected through a disadvantaged link. The proxy can even cache results for subsequent service calls.

There is a problem related to signature values. Equivalent POJO and SOAP messages will have different signature values, and the integrity of the message is broken during a conversion. The proxy can sign the converted object using its own private key, which would require that the service accepts that the proxy vouches for the original client in the authentication phase.

VIII. SOAP Guard and Confidentiality Labeling

As can be seen in Figure 1, a SOAP guard connects military networks of different classification levels as an application gateway in the form of an HTTP proxy. It relies on *confidentiality labels* that are bound to information object in a form that can be inspected and validated by the guard in order to make decisions whether to allow objects to be transferred from a high to a low classified network. The transport may be initiated by a client on the low side as an HTTP operation (e.g. a Web Services request), in which case the response will need a label in order to pass through. The request will need to be labeled if it is initiated on the high side.

The format requirements of the label is expressed in a proposed NATO standard for information labeling [10], and describes the structure of the label, the signature and the binding mechanism.

The proposed standard does not mandate the validation of labels, but implies that there must be a PKI-type certificate validation process in order to trust the validity of a label.

Nor does the NATO standard set requirements to the labeling process. In order to provide a trusted label, the process of creating and attaching a label must be robust against attacks from malware etc, and should be executed with high assurance.

IX. Challenges and potential problems

This section reports some of the technological problems that were observed during the configuration and pre-testing of the experimental set-up:

A. Android client and IPSec

The IPSec client on the Android platform is a basic implementation for connection to Microsoft IPSec services, which means that it only supports IPv4, IKEv1 keying protocol and relies on the use of L2TP and PPP protocols on top of the IPSec connection. The entire CoNSIS experiment was based on the use of IPv6, but the Android link required a different configuration. There is general support for IPv6 in Android, but the kernel is not able to manually set the IPv6 address of an interface, which makes a tunneling arrangement infeasible.

The Android IPSec appeared to to use an inactivity timer to disconnect an idle link. This was not welcome over an XMPP connection that carried infrequent messages.

B. XMPP as a messaging service

Although XMPP messages can carry any data and connect to any client, it was not ideal as a message service. The XMPP standard has chat messages in mind, and mechanisms related to presence, file transfer, avatars, rosters etc. were prominently implemented in the XMPP server. In particular, the facility to store messages that could not be delivered due to offline clients were not welcome in a messaging system used for request/response traffic. The final choice of XMPP server (OpenFire) offered an option to discard such packet, which improved its utility greatly. This server also offered centrally managed rosters, which relieved the client from creating the rosters themselves.

The XMPP standard offers extensions for PubSub communication (XEP-0060), which is potentially a good candidate for the transportation of service invocation messages.

OpenFire implements the PubSub extension, but without any administrative tools (management of nodes and subscriptions etc.). Without such tools, experimentation on PubSub messages becomes very tedious.

The XMPP connections rely on stable IP routes in the network. For purposes of chat application in tactical networks, studies has been conducted to distribute message through diffusion or gossip techniques [11]. Future experiments could possibly pursue those opportunities.

C. Android network routing

Although Android has several networking interfaces (WiFi and 3G) and contains a Linux kernel, it does not appear to offer routing to these interfaces. All network traffic is sent to the WiFi adapter if the link is up, otherwise the 3G service is used. One initial idea was that the Android unit could access NGO resources (i.e. the Identity Provider) over a 3G connection and military resources over the WiFi/IPSec connection.

Without that option, the NGO resources (the IdP) had to be placed in the military network (as seen on Figure 1). This was far from an ideal situation and was not intended in the early experiment design.

X. The CoNSIS evaluation

The background for the efforts presented in this paper is the collaboration program called "Coalition Network Secure Information Sharing" (CoNSIS), with participation of military and industrial scientists from Germany, France, USA and Norway. The program was operational from 2010 and its objective is "to develop, implement, test and demonstrate technologies and methods that will facilitate the participants' abilities to share information and services securely in ad-hoc coalitions, and between military and civil communication systems, within the communications constraints of mobile tactical forces".

Another objective of CoNSIS is that "The participants intend to utilize, to the maximum extent possible, commercial standards to minimize interoperability difficulties. Only those elements of the technical architecture which are not available from the open market will be investigated, and potentially developed."

The main deliverance of the CoNSIS program is a technical test and demonstration which took place in Greding, Germany, during June 2012. During this demonstration, communication spanned vehicles from several countries and a number of national headquarters, using different radio systems and security technologies to access services and to exchange information. The technology experiment presented in this paper is only one of large set of experiments which took place.

XI. Conclusion

This part of the CoNSIS experiment was conducted with the intention to study a range of security technologies for the separation of military and civilian networks, and to study how commercial mobile units (a waterproof Android smartphone) could be employed inside that security framework.

Most of the technologies (StrongSwan IPSec, serialized Java objects, homemade IdM, SOAP Guard) was working well. The use of Android was a bit over-ambitious, in the sense that IPv6, IPSec and network routing was implemented in a rather basic fashion.

The Android unit turned out to offer excellent portability of existing Java SE sources, and the XMPP stack was directly ported to Android without the need for any corrections. The low price, availability of development tools and the existence of waterproof Android units is promising for the future use of mobile COTS units in tactical networks.

References

[1] R. M. Zich, "Warfighters and humanitarians: Integrating technology to save lives," 1997. [Online]. Available: http://www.globalsecurity.org/military/library/report/1997/Zich.htm [Retrieved Apr 30, 2012]

[2] A. Fongen, "Federated identity management for android," in *SECURWARE 2011*. Nice, France: IARIA, July 2011.

[3] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST model for role-based access control: towards a unified standard," in *RBAC '00: Proceedings of the fifth ACM workshop on Role-based access control*. New York, NY, USA: ACM, 2000, pp. 47–63.

[4] A. Fongen, "Optimization of a public key infrastructure," in *IEEE MILCOM*, Baltimore, MD, USA, Nov. 2011.

[5] "Shibboleth." [Online]. Available: http://shibboleth.internet2.edu/ [retrieved November 9, 2010]

[6] "OpenID." [Online]. Available: http://openid.net/ [retrieved November 9, 2010]

[7] "The Libery Alliance." [Online]. Available: http://www.projectliberty.org/ [retrieved November 9, 2010]

[8] A. Fongen, "Identity management without revocation," in *SECURWARE 2010*. Mestre, Italy: IARIA, July 2010.

[9] ——, "Architecture patterns for a ubiquitous identity management system," in *ICONS 2011*. Saint Maartens: IARIA, Jan. 2011.

[10] S. Oudkerk, I. Bryant, A. Eggen, and R. Haakseth, "A proposal for an xml confidentiality label syntax and binding of metadata to data obejcts," in *NATO RTO Information Technology Panel Symposium, Information Assurance and Cyber Defence*, Antalya, Tyrkia, 2010.

[11] M. Skjegstad, K. Lund, E. Skjervold, and F. T. Johnsen, "Distributed chat in dynamic networks," in *IEEE MILCOM*, Baltimore, MD, USA, Nov. 2011.