PROPOSAL FOR AN INTERNATIONAL RADIO SECURITY SERVICE API FOR TACTICAL RADIOS

Scott Leubner (Harris Corporation, Rochester, NY; sleubner@harris.com); Anthony DiBernardo (Harris Corporation, Rochester, NY; adiberna@harris.com); Charles Linn (Harris Corporation, Rochester, NY; clinn@harris.com); Leonard Picone (Harris Corporation, Rochester, NY; lpicone@harris.com); Rafael Aguado Muñoz (Indra, Aranjuez, Madrid, Spain; ramunoz@indra.es); Javier Fernandez Alonso (Indra, Aranjuez, Madrid, Spain; jfalonso@indra.es); Alvaro Mayol Garrido (Indra, Aranjuez, Madrid, Spain; amayol@indra.es)

ABSTRACT

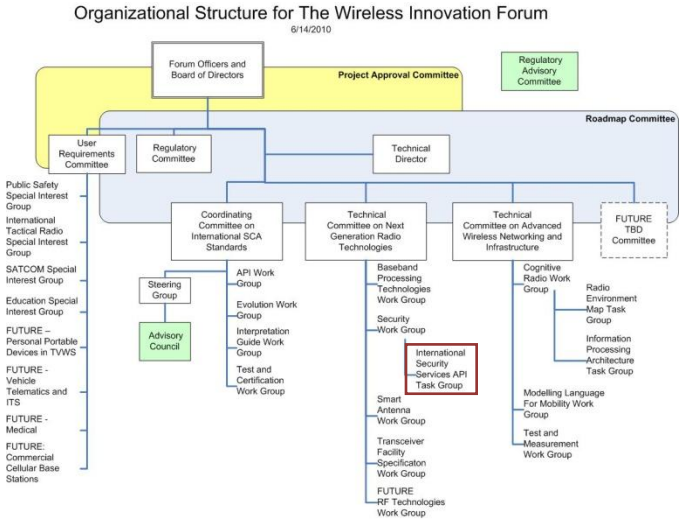
In 2001, the JPEO JTRS established the landmark SCA v2.2, formally defining what would become the military communication standard. The SCA specification defined for the first time what would become the basis of military radio terminals and the pillar architecture to support waveform development. The primary objective of this specification was waveform portability enhancement, allowing independent radio manufacturers to migrate waveforms developed by other vendors onto their platforms. However, over time, the inclusion of export-restricted elements in the Security Supplement impeded the standard's international acceptance. In this context, the Wireless Innovation Forum (WInnF, formerly called the SDR Forum) initiated an International Radio Security Service Application Programming Interface (IRSSAPI) working group. Over the past year, this group has been devoted to the development of a truly international Security API standard for radio communications with no export restrictions. In addition, this group has and will continue to act as a catalyst between the WInnF development community and government stakeholders. The paper will show the basis upon which the Security Services API will be based, as well as the main drivers for its definition. The paper will conclude with the main achievements and the expected future work.

1. INTRODUCTION

The JPEO JTRS released its initial version of the Software Communications Architecture (SCA) with the basic goal of standardizing the operation environment (OE) for software definable radio systems. As the standard gets more mature, the initial set of goals was refined to fulfill the market requirements, focusing on the facilitation of the portability of waveforms between different platforms. In order to achieve these set of goals, the earlier versions of the SCA included both an API supplement [3] and a security supplement [4] to define the applicable system APIs between waveform components and the OE and between OE components themselves.

However, later revisions of the SCA deprecated both these supplements to instead give preference to API standards being developed by an API standardization committee. Unfortunately, today there does not exist an internationally available API standard that defines the security interfaces for SCA based radio systems. Recognizing this gap, the WInnF has endeavored to create a security services API, applicable to the international community, for standardizing the interfaces of the radio security services (RSS) provided by an SCA based radio platform.

Established in 1996, the Wireless Innovation Forum™ is a non-profit mutual benefit corporation dedicated to driving technology innovation in commercial, civil, and defense communications worldwide [7]. The Forum is composed of five different committees, User requirements, Regulatory, International SCA Standards, Next Generation Radio Technologies and Advanced Wireless Networking. As part of it structure the IRSS API Task group is integrated into the Technical committee on Next Generation Radio Technologies as a task group of the Security Work Group. The following picture shows the IRSS API Task group as part of the WInnF structure:



Although the access to the task group web portal is restricted to WInnF members only, the call for participation is open and can be found here [9].

THIS INFORMATION WAS APPROVED FOR PUBLISHING PER THE ITAR AS "Fundamental Research".

Following the aforementioned general SCA goals, the objective of this API, called the International Radio Security Services API (IRSS API), is to extend the waveform portability between different SCA based platforms to the security boundary. By standardizing the security API, the WInnF's IRSS API task group promotes portability of waveforms developed against those standards to platforms that provide those APIs. The following picture provides a brief overview of how the inclusion of this API fills the missing piece in the portability puzzle:

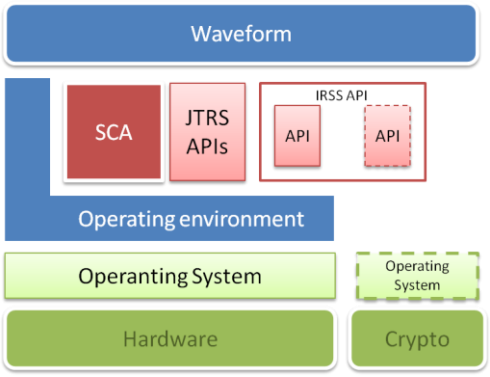


Figure 1. IRSS API localization

As it is shown in the picture, the IRSS API is a large service that has to support waveform components in their interaction with the platform but also has to provide support among the platform components. Therefore, it is essential to specify the interfaces used by the waveforms as those interfaces foster portability. On the other hand, waveforms do not connect to and use the security interfaces provided for other platform components, and thus, specifying platform security interfaces only serves to constrain platform development without adding to waveform portability. Taking into account all these considerations the IRSS API focuses on detailing security interfaces that are likely used by waveforms.

To develop this API, the IRSS API task group drew upon its experience with existing waveforms and on existing security APIs. In particular, the working group considered use cases for legacy circuit-based waveforms and also newer networking waveforms. Existing security APIs referenced include version 1.1 of the deprecated Security Supplement to the SCA [4], which defined the original RSS API for SCA based systems, and the Common Interface to Cryptographic Modules (CICM) [5], which is an IETF draft to standardize interfaces to cryptographic modules.

2. DESIGN STRATEGIES

The first task entrusted to the IRSS Task group was to define the strategy to be followed for the design of the API. The software design has always by divided into two opposite tendencies, depending in the decision of when the code writing has to start. Those two extreme are composed of proponents of a wide design stage, because is easier to solve errors at an earlier stage and those defending that a wide design phase can end in paralysis analysis.

Inside the multiple options that the current software design trends, there are two basic design strategies that can be followed:

- Top – Down. The top – down approach is also known as stepwise design. The basis of this strategy is the decomposition of the system in several subsystems, avoiding entering in too many details. Then a similar strategy is followed with all the found subsystems. The group requested the help of the International Tactical Radio SIG to identify the most relevant waveforms to be taken into account during the specification of the IRSS API. These waveforms will be the basis of the use case definitions that will drive the specification of the IRSS API.
- Bottom – Up. The bottom – up strategy is also known as synthesis, in which the system is built on the basis of the identification of the basic components, and their interactions among them. The group started to identify some components that will be intended as the basic pillars of the IRSS API specification.

To streamline the overall project schedule, it was decided to have a mix approach in which both the IRSS task group and the International Tactical Radio SIG started the work simultaneously. The use cases defined in the top – down approach are being used to refine the components found in the bottom – up work.

3. API GUIDING PRINCIPLES

Before launch the definition process of the IRSS API is essential to define the development and design principles that will guide the IRSS API specification. To this end, the following IRSS API development guidelines were identified.

4. SCALABILITY

Due to its own nature, as waveform driven definition, scalability is of paramount importance for the IRSS API task group. The API needs to accommodate future growth since radio security services will continue to evolve, as the waveforms are not a static universe. Also, although the API is intended to ease portability of waveforms between platforms, it should not unduly constrain those platforms to prohibit sovereign waveform development. In order to garner acceptance of the API in an international community, the interface must allow for scalability.

This leads to a follow-on concept: capability negotiation. Extensible APIs imply that there will be an evolution of the API as technology progresses. This further implies that multiple revisions of the IRSS API will exist. The IRSS Task Group should consider building in a capability for a waveform to query/negotiate the capabilities of the IRSS that it interfaces to.

4.1 Adaptability

The WInnF, as it was shown along the paper, intends the IRSS API to be platform agnostic, being possible to use the API in different SCA based platform. As such, the IRSS API needs to be transformable to existing RSS APIs or to platform unique CSS APIs. To foster broad international acceptance, since these platforms will use different crypto sub-systems, the IRSS API task group should not develop the API with any particular CSS in mind as that might constrain the selection of a CSS. Likewise, implementation details should be avoided in the definition of the API as this would constrain the transformation options.

4.2 Export Restrictions

To facilitate international acceptance and public distribution, the IRSS task group needs to make the distribution of the IRSS API free from export restrictions. (e.g. it must be free from ITAR restrictions).

4.3 Consistency

The Software Communications Architecture and the publicly available JPEO JTRS API standards (see references in section 1) have generally been embraced by the international Software Defined Radio (SDR) community. As such, there are a number of platforms that have been or are being developed against these standards and WInnF expects that SDR developers will host the IRSS API on these platforms. This does not preclude the use of other SDR architectures, but the IRSS TG would be remiss not to tailor to these standards. In addition, WInnF has opted to use the format of the JPEO JTRS public APIs in other API TGs and the IRSS TG should do the same. Finally, the use of Interface Description Language (IDL) allows for a language independent specification of the API and continues to foster deployment in SCA-based systems.

4.4 State - Flow Architecture

Many software subsystems utilize a state machine to control the execution and behavior of the subsystem. In many cases these states are implied, or even exposed, through a client API. The radio security service is no exception to this and may itself employ a state machine. As such, the details of these states, as they relate to client usage of the API, need to be communicated to the service users. Further, the flow of events dictates the semantic usage of the API and drive the underlying state machine. Understanding the event flow alternatives becomes essential to understanding the use of the API itself, and thusly, the IRSS API needs to communicate this semantic behavior to service users as well.

4.5 Good Practices

The IRSS API is an interface specification, as such, the IRSS task group need to use good practices when specifying this API. Those best practices are taken from various well known software engineering technologies. The main source of these practices comes from Object Oriented (OO) design practices used in defining class interfaces. The OO defines primitive operations as "those that can be efficiently implemented only if given access to the underlying representation of the abstraction" (Booch, 1994). In contrast, a non-primitive operation is one that can be implemented using other primitive operations. There are other definitions that provide some additional interface guidance by emphasizing complete, yet minimal interfaces (Meyers, 1998). An interface is complete if it sufficiently supports all capabilities that a client would reasonably expect from the abstraction. In contrast, a minimal interface provides as few operations as possible without having any overlap in functionality between those operations.

These good practices have to be translated also to the interface definition. For example, the order of parameters for operations with similar signatures should be consistent and the names of those parameters should be consistent. Also, the IRSS TG should employ a consistent exception philosophy for those operations that return exceptions.

4.6 Separation of Concerns

The standardization of the SCA allowed radio manufacturers the implementation of a truly platform architecture, decoupling the pure waveform development from the specific platform. This concept has evolved until today's separation of concerns paradigm, where the development of the waveform and the platform run over different paths.

The goal of the IRSS API is to promote waveform portability between differing platforms. By standardizing the security API, the IRSS TG promotes portability of waveforms developed against those standards to platforms that provide those APIs. Thus, it is essential to specify the interfaces used by the waveforms as those interfaces foster portability. On the other hand, waveforms do not connect to and use platform interfaces, and thus, specifying platform interfaces only serves to constrain platform development without adding to waveform portability. Other separation of concerns considerations include separating interfaces in support of least privileges principles.

The previous discussion mentions a couple considerations, but there are additional ways to divvy up functionality between interfaces that the IRSS TG can consider. For example, separating real-time functionality from non-real-time functionality may provide some additional benefit. In the end, the IRSS TG should sufficiently analyze the resulting interface structure to insure a proper separation of concerns,

4.7 SWaP

The overall trend in communications technology is smaller, yet faster and more powerful. These trends are often at odds however. Smaller, battery-powered systems (commonly referred to as size, weight, and power (SWAP) constrained systems) have less processing horsepower and processing capabilities than their full size counterparts, yet they are expected to perform at equally high levels. The way to enable these systems is through intelligent, efficient programming. With this in mind, the IRSS TG must define interfaces that are efficient to support SWAP constrained systems. Careful

consideration must be given to their use in time-critical/data flow areas of the API.

- One area to avoid is the use of CORBA::Any's. CORBA::Any's carry a lot of overhead with them making their use inefficient.
- The other area to consider is the use of two-way function invocations. Two-way calls, though sometimes necessary, are less efficient than their one-way cohorts. Careful consideration must be given to their use in time-critical/data flow areas of the API.

4.8 Security Design Specific

Waveforms will connect to and use the capabilities provided by the security service; however, a system should only give a waveform access to those services that it requires. By grouping capabilities into distinct functional interfaces and restricting waveform usage of those interfaces through security policy enforcement, a system can limit what services a waveform has access to, thus enhancing overall security.

4.9 Complexity

To promote acceptance in the international community, the IRSS TG needs to specify an API with an understandable level of complexity. Key to this is providing clear, concise documentation with detailed diagrams to convey the intended use and semantics of the interfaces. Ambiguous or obscure verbiage should be avoided. Furthermore, the structure and definition of the interfaces should promote their ease-of-use and understandability. Consistency helps to promote understandability.

5. CATEGORIES OF SECURITY APIS

The first EDA SDR Conference, hold in Finland in November 2009[6] defined the SDR standardization process in different terms depending on the level of classification of the information to be standardized. The standardization process should be divided in three different areas, international, multinational and individual, depending on the actors involved in the standardization process. This statement was officially endorsed by the Wireless Innovation Forum in February 2010.

Although the security API must be designed in such a way that application portability can be achieved, still is necessary to protect coalition and sovereign country interests. In general, APIs pertaining to security can be classified into the three categories mentioned before: internationally open standards, multi-national standards, and, the most restrictive, sovereign standards.

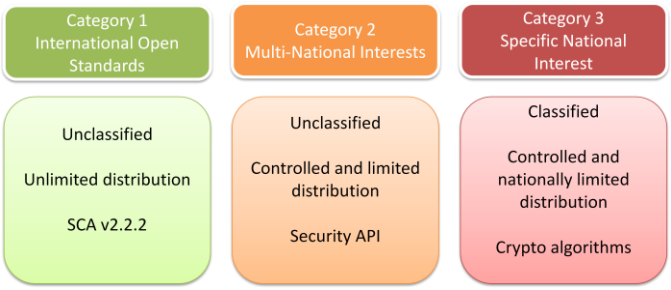


Figure 2 - Categories of Security APIS

This model details separation of public, coalition and sovereign categories. The goal is to allow freedom to employ proprietary/sovereign content without changing the public API. Sovereign standards are developed by nations to serve their national interests. They are by definition limited in distribution, tightly controlled by government export restrictions. An example of this would be interface control documents for US Type-1 cryptographic security ASICs. Examples of sovereign and coalition materials include:

- Specific encryption algorithms, formats, initialization and usage
- Specific crypto algorithm structure
- Key formats, tagging
- Crypto initialization sequences
- Details on crypto outputs, states, etc.
- Limits, bounds for crypto bypass
- TRANSEC formats and parameters
- Specific means of providing access to critical functions (e.g. zeroize)
- Authentication keys, etc.
- Specific format of authentication material (Key Management Infrastructure (KMI) etc.)

Multinational standards loosen distribution limitations to some extent, as nations look to support and collaborate with their allies. The international nature of the API requires that in addition to public algorithms, coalition or sovereign national algorithms must be supported. Some details of these algorithms cannot be openly published, but must still be supported in the context of the IRSS. The API must therefore be defined in sufficiently generic terms to allow for the implementation of these country-specific security capabilities. In cases where it is not possible to be completely generic, the transformation layer can be employed to arbitrate between the API layers (see next section). The IRSSAPI must be defined in such a way as to keep the need for API refactoring and/or IDL extensions to an absolute minimum.

6. TRANSFORMATION LAYER

The previous chapters have introduced not only the most important drivers for the definition of the IRSS API interface, but also the need for the adequacy of the standard to different degrees of the classification of the information.

The most widely accepted technique to protect classified information is the use of a Transformation Layer. This layer will keep the portability among different platforms while at the same time prevents the leak of information to non-authorized actors. The process of porting a waveform from one platform to another must take into consideration the following consideration:

- Required Information Assurance (IA) and security capabilities. Every waveform can have different degrees of requirements on what regards Information Assurance and security capabilities. Generally each radio manufacturer or even each country may have different ways to provide the mechanisms to assure the adequate levels of security.
- Waveform functional partitioning across security boundaries.
- Policy based re-configurability. The use of policies is one of the basic mechanisms to reconfigure the security system. The application of these policies has to be supported by the platform to which the waveform is being ported.
- Connection to open networks. During the porting of a Waveform it is important to define how it will be connected to other networks, the kind of the networks, protocols, etc.

The need to take into account all these factors is of capital importance to define a multi-surface model that will facilitate both open API standardization (IRSS API) and protected interests or private API.

Transformation layer used to uniquely and securely alter information format and content. Therefore the National or coalition APIs only exposed where appropriate.

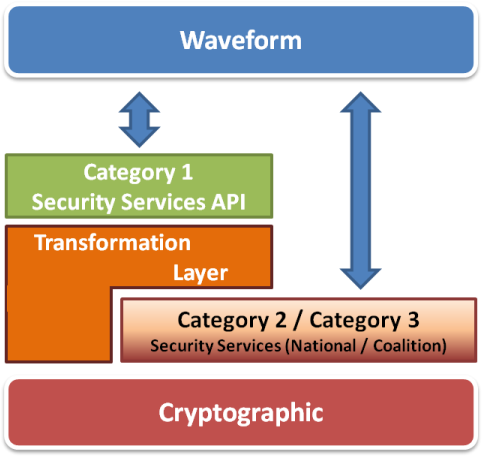


Figure 3 - Waveform / Security API Context

7. API ALTERNATIVES

Given the criteria and motives set forth above in selecting an appropriate API, there are a handful of options available to the radio developer.

7.1 SCA

The selection of the SCA to implement the radio security services is the obvious choice for API definition, as long as export restrictions are not a concern. However, there are some considerations that make unfeasible the choice of the SCA as the security reference API. Yet, the security supplement can be taken as reference work for future developments.

The security supplement has been evolved into its version 1.1 as part of the JTRS SCA 3.0, but the whole release has been unsupported, making difficult to use it as reference. Despite these considerations, there are a number of proponents that defend this document as the basis for future secure services development.

7.2 Common Interface to Cryptographic Modules (CICM)

The CICM specification has been developed recently to satisfy the need for an international security API. Currently in draft form in the IETF, the specification does not face the export concern that impacts the SCA. However, there are a few points that make the selection of CICM as the API of choice slightly less than obvious –

- Traffic concerns – one packet per CORBA call, byte-oriented methods, blocking traffic flow, two calls for processing of in-band bypass.
- There is no obvious "black side" component to CICM. All interface definitions are applicable to red side only.
- It is a complex interface, with multiple levels of inheritance. Complexity directly impacts portability and adoption rates.

7.3 Custom

A third choice is for radio developers to create their own custom APIs independently. This simplifies the task of development by removing the dependency on external standards. It also enables sovereign nations to develop their own security APIs. The downside is obviously portability. Having a standard API at the cryptographic level is essential to enable efficient waveform porting. Platforms are still able to implement country- or entity-specific security APIs as Category 2 or 3 APIs with this approach, as long as they also develop to a standard API and incorporate a transformation layer.

7.4 The International Radio Security Service API

The fourth option, and the one to which the Wireless Innovation Forum has dedicated its resources, is to develop and internationally agreed upon API that is not limited by export restrictions and leaves sufficient flexibility to enable

sovereign interests to be maintained. This API is summarized in the following section and explained in much more detail in the companion paper, "A Technical Overview of the International Radio Security Service API for Tactical Radios".

8. SUMMARY OF MODULES

With the focus being on waveform portability, a handful of obvious security concepts are not included in the IRSSAPI. Those include APIs related to key fill, user administration, algorithm management, portions of key management, audit, certificate management, etc.

8.1 The Control Module

The Control module contains interfaces related to waveform configuration and control of the security services. These include the ChannelMgmt interface and the CertificateMgmt interface.

8.2 The Infosec Module

The Infosec module contains interfaces that waveforms use to access the information security services of the radio. These services fall into two categories: transformation services and TRANSEC services.

8.3 The Bypass Module

The Bypass module contains interfaces that waveforms utilize to bypass control messages through the cryptographic subsystem. The IRSS provides bypass support through a pair of interfaces, one provided by the security subsystem and one provided by the waveform.

8.4 The IandA Module

The IandA module defines interfaces that waveforms use to access the I&A features of the security subsystem. These features include generating hashes, generating message authentication codes (MAC's), and generating and validating signatures.

8.5 The Protocol Module

The Protocol module defines interfaces that waveforms use to exchange protocol messages with the security subsystem as part of an asymmetric key protocol. These interfaces define a generic messaging protocol that supports the various message exchanges needed by different protocols.

8.6 API Specification

These modules have been defined in the IRSSAPI specification, available through the Wireless Innovation Forum. The "Specification for the International Radio Security Service API, v1.0" (assumed, pending forum approval) can be found at http://www.wirelessinnovation.org/. (A more complete URL directing readers to the actual document will be provided once the specification is available to the public.)

9. CONCLUSIONS

The biggest achievement in the current SDR development panorama is to increase the efficiency of the portability effort, to truly streamline the cost of the migration of a given waveform from one platform to another.

The publication of the SCA and the successive initiatives (e.g. ESSOR) have improved enormously the portability of the existing platforms. However is also true that security has traditionally become a burden making impossible to achieve a convergence in the actual SDR platforms, increasing not only the complexity of the migration of the waveform but also the overall interoperability of the terminals.

The paper has shown the problematic around portability issue on the SDR panorama, proposing the missing piece to achieve a truly portable platform. Not only the basic principles and design drivers has been presented but also a brief presentation of the modules composing the API specification. The WInnF IRSS API task group has achieved also the implantation of the different levels of classification model. The basis of this model is a mechanism that is able to keep the portability while at the same time preserve the security of the overall system and the integrity of the sovereign or coalition API.

[1] Modular Software-programmable Radio Consortium, *Software Communications Architecture Specification*, MSRC-5000SCA, v1.0, May 17, 2000

[2] Modular Software-programmable Radio Consortium, *Software Communications Architecture Specification*, MSRC-5000SCA, v2.2, November 17, 2001

[3] Modular Software-programmable Radio Consortium, *Application Program Interface Supplement to the Software Communications Architecture Specification*, MSRC-5000API, v1.1, November 17, 2001

[4] Modular Software-programmable Radio Consortium, *Security Supplement to the Software Communications Architecture Specification*, MSRC-5000SEC, v1.1, November 17, 2001

[5] D. Lanz, L. Novikov, *Common Interface to Cryptographic Modules (CICM)*, Internet Engineering Task Force, Internet-Draft, January 7, 2011

[6] EDA SDR Standardization process, http://www.eda.europa.eu/genericitem.aspx?Area=Organisation&ID=594, November, 2009

[7] www.wirelessinnovation.org

[8] Support for "Three Category" Approach for Software Communications Architecture (SCA) Standards and Extensions, WINNF-09-R-0023-V1.0.0, February, 10, 2010

[9] IRSS API Call for participation, http://data.memberclicks.com/site/sdf/Project_IntlSDRSecurityServicesAPI.pdf