On Secure Mobile Communication Software Issues

Manfred Sneps-Sneppe, Dmitry Namiot

***Abstract*— The paper considers some secure mobile communication software issues from both the tactical and strategic points of view and some failures due to overly complex software as well as cyber threats. The failed tactical communication systems include Future Combat Systems (due to software issues), and Joint Tactical Radio System (after failed integration tests). Joint regional security stacks (JRSS) are the main components of the cybersecurity structure. In 2018, the Pentagon suspended this $2 billion cybersecurity project due to unsolved software issues. Some myths about CORBA and Software Communications Architecture (SCA) are discussed and some unsolved software issues are pointed out. This paper considers the experience of using software-defined radio systems (SDR) in special systems. First of all, we are talking about the Future Combat Systems (FCS) project and a software-defined radio (SDR) system - a radio communication system that uses software to process various signals.**

Keywords—software-defined radio, CORBA, Software Communications Architecture, Defense Red Switch Network

I. Introduction

The paper considers some secure mobile communication software issues from both the tactical and strategic points of view and some failures due to overly complex software as well as cyber threats. The area considered is a hot one: since 2022 only, Google Scholar gives 13,300 results on call “SDR software”, 16,700 results on call “Software Communications Architecture” and 16,800 results on call “secure mobile communication software”. Therefore, it is hopeless to survey the volume of research in the world and limit ourselves to a narrower study.

This paper considers the experience of using software-defined radio systems (SDR) in special systems. First of all, we are talking about the Future Combat Systems (FCS) project and a software-defined radio (SDR) system - a radio communication system that uses software to process various signals (modulation, demodulation, decoding, etc.) [1]. With all this, in general, the experience of using SDR in special applications, today, should be recognized as not very successful. And the reasons for this, in our opinion, lie precisely in the software. Programs are now rarely created from scratch, most often it is about using any frameworks. This is especially true for distributed systems, which, by definition, are radio systems. SDR implementations were no exception in this regard, and OMG CORBA [2] was considered as system software. CORBA (Common Object Request Broker Architecture) was introduced in the 1990s, and was created for the portability of remote procedure call (RPC) applications in distributed systems. Over time, CORBA, due to its complexity and bulkiness, has practically disappeared from the development world. But this was not at all the case in corporate and special systems. As a result, the architecture of the systems turned out to be tied to outdated software components, for which the task of finding developers became the number one problem.

What about Strategic communications, they, first, relate to the Defense Information System Network (DISN) cybersecurity. Joint regional security stacks (JRSS) are the main components of the cybersecurity structure to protect computers and networks in all military organizations. Unfortunately, in 2018, the Pentagon suspended this $2 billion cybersecurity project due to unsolved software issues. In conditions of cyber war, the Defense Red Switch Network (DRSN) for top secret communications uses 40 years old, but secure ISDN technology. The Secure Communications Interoperability Protocol (SCIP) is developed for circuit-switched one-to-one connections (including GSM), not packet-switched networks. Thus, the use of SCIP in packet radio networks is unclear till now.

This work is devoted to the current state of such systems from both tactical and strategic points of view, their software architectures, and possible directions of development. The remainder of the article is structured as follows. In section II, we describe tactical communications. Section III deals with strategic communications.

II. On Tactical Communications

A. Future Combat Systems failure due to software (2003-2009)

FCS was the largest and most ambitious program in US Army history [3]. The FCS program ran from 2003 to 2009 and was canceled a year later due to software problems.

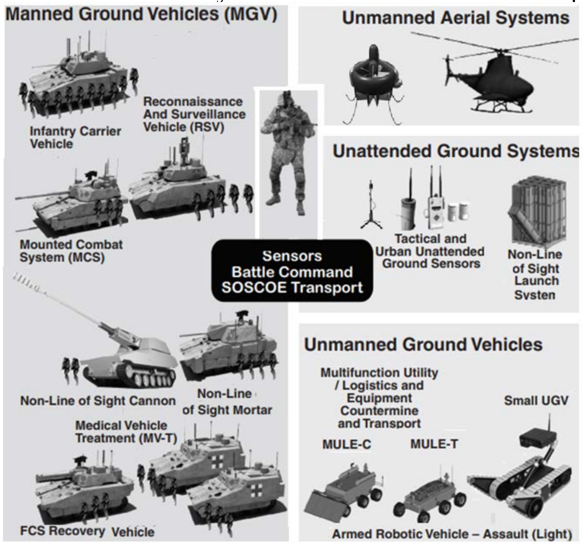
Manned Ground Vehicles (MGV)
Infantry Carrier Vehicle
Reconnaissance And Surveillance Vehicle (RSV)
Mounted Combat System (MCS)
Non-Line of Sight Cannon
Non-Line of Sight Mortar
Medical Vehicle Treatment (MV-T)
FCS Recovery Vehicle
Unmanned Aerial Systems
Unattended Ground Systems
Tactical and Urban Unattended Ground Sensors
Non-Line of Sight Launch Svsten
Sensors Battle Command SOSCOE Transport
Unmanned Ground Vehicles
Multifunction Utility / Logistics and Equipment Countermine and Transport
MULE-C
MULE-T
Small UGV
Armed Robotic Vehicle – Assault (Light)

Fig. 1: Basic 18 FCS systems [3]

The volume of FCS software under development was estimated at 60+ million lines of code. FCS was based on its own operating system - SOSCOE. SOSCOE was conceived as a network-centric operating system integrating individual FCS communication software packages (Fig. 2). This solution required the development of more than 100 programming interfaces for external applications.

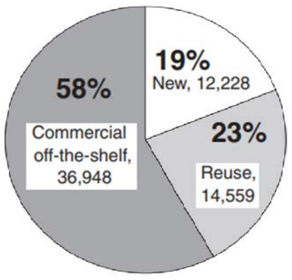

Fig. 2: FCS Projected Software [3]

FCS was the US Army's main modernization program from 2003 to early 2009. The US Army said it was their "most ambitious and far-reaching modernization program" since World War II. Spending amounted to more than $30 billion with no significant effect, in 2009, the program was announced to be stopped [4].

B. Joint tactical radio system failure due to software (1997–2011): CORBA is dead

The Joint Tactical Radio System (JTRS) played an important liaison role in the FCS program. JTRS is a programmable, broadband, open architecture secure communications system. It provides virtual circuit and datagram services. This ensures reliable, simultaneous, multi-channel transmission of voice, data, images, and video. Theoretically, digital signal processing provides flexibility and ease of operation in end-to-end communications, packet formatting, and packet switching protocols.

JTRS was supposed to replace about 30 different military systems and become a universal link for all weapons systems (combat vehicles, manned aircraft, drones, missiles, etc.), which would be combined into a single control system. In this case, the JTRS devices had to act as a telephone, a computer, and a network router at the same time. JTRS was an attempt to unify military radios through digital signal processing. Technically, JTRS is a collection of software-defined radios. The amount of code is more than four million lines of code. JTRS was launched in 1997 and canceled in 2011 after failing integration tests [5].

In testing, it was noted that some radios took more than 10 minutes to boot up and become operational when they could transmit or receive. Obviously, delays are simply unacceptable. JTRS is widely seen as one of the DoD's greatest acquisition failures, having spent $6B over 15 years without delivering a radio [6]. This, in fact, led to the fact that the concept of network-centric warfare was never implemented. Currently, there is a turn towards decision-oriented warfare [7] based on artificial intelligence. The new concept is also software-based, even to a greater extent. Will it be more successful?

Again, for compatibility with existing systems, the implementation of many software adapters was necessary. The general consensus was that CORBA, in addition to slowing down development, was also a slow product. This is refuted in [8], but with a very peculiar argument: CORBA is designed with portability in mind. However, let's look at it from the side of the SDR developer. Now there are simply no ready-made components that could be transferred. They just don't get developed. Accordingly, what is the benefit for developers? What justifies the expense of learning a complex specification, long development time, and more (because of complexity) bugs? Many CORBA APIs are much larger than necessary. The CORBA object adapter requires hundreds of lines of interface definitions, although the same functionality can be implemented in about a dozen lines, and the rest is a description of how the program interacts with the CORBA runtime.

At this point in time, the idea of using CORBA looked hopeless. But then Software Communications Architecture (SCA) appeared.

What is left of the ambitious JTRS project? JTRS HMS (this abbreviation means Handheld, Manpack & Small Form-Fit) radios, for use by the individual solder, are now headed into production. The AN/PRC-154A Rifleman Radio is designed to be carried by leaders needing secret access to the platoon [9]. This is a personal radio for use directly on the battlefield. JTRS HMS AN/PRC-154 Rifleman radios weigh less than a kilogram (with a 10-hour battery and antenna) that can create self-forming ad hoc networks for voice and data transmissions (Fig. 3). These SDRs can securely transmit voice and data simultaneously using Type 2 cryptography and the new Soldier Radio Waveform.

Fig. 3: AN/PRC-154A Rifleman Radio [9]: three antennas for three frequency bands but not a single software-defined antenna as planned [4]

C. The revisited SDR concept (from 2014): on CORBA and SCA myths

Software Communications Architecture (SCA) presents the so-called Operating Environment (OE) for waveform applications and allows some separation between waveform applications and radio platforms (Fig. 4). The OE enables to manage of waveform applications and provides radio platform services (transceiver, modem, etc.). The idea behind SDR is that there is a digital signal processor (DSP) that will perform the tasks of encoding and decoding, modulation and demodulation, timing, and signal processing. The DSP module consists of many programmable digital circuits. The latter may include an application-specific integrated circuit (ASIC), a field programmable gate array (FPGA), a conventional processor (in some articles, general purpose processor or GPP), or a combination of these elements.

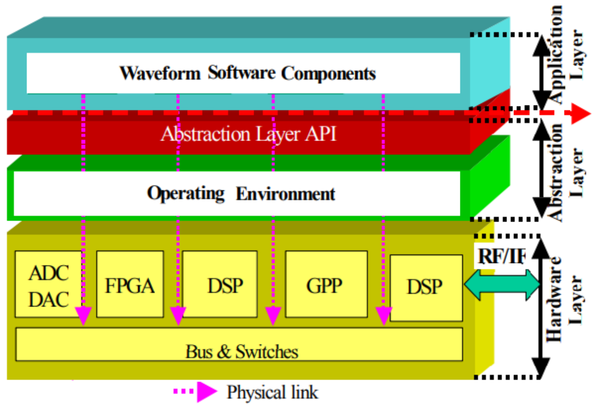

Fig. 4: Distributed SDR architecture [10]

SCA defines a standard framework for instantiating, configuring, and managing software. Tactical Radio Application Programming Interfaces (APIs) present a suite of interface specifications. Each of them defines a key software interface for an abstraction of the underlying product-specific software functionality or physical hardware. The SCA and APIs improve cyber security and performance, promote competition and reduce lifecycle costs. Fig. 5 illustrates SCA Architecture Layer Diagram.

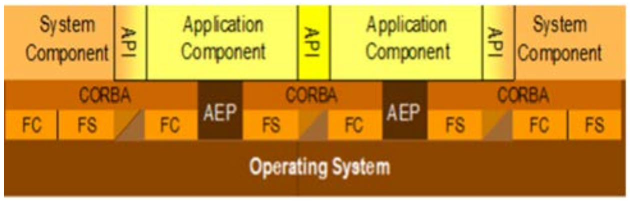

Fig. 5: SCA Architecture Layer Diagram

The software components which provide for the management and execution of the SCA applications and devices comprise the SCA-defined operating environment (OE). The OE consists of an operating system (OS), CORBA middleware (including the OMG-defined Event SCA and Naming Services), and the elements defined by the Framework Control (FC) and Framework Service (FS) Interfaces, AEP=SCA Application Environment Profile. SCA supports domain-specific APIs and the JTRS Radio-specific APIs; they are called the JTNC APIs (Fig.6).

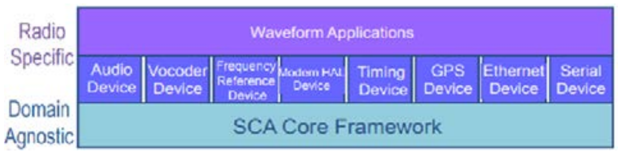

Fig. 6: JNTC API [8]

The work [8] from Nordiasoft already mentioned above is devoted to the protection of SCA. The thing is that, as can be seen from Figure 6, this is some kind of shell (add-on) on the same CORBA. If CORBA was criticized all the time for being slow, then the add-on, for obvious reasons, could not get faster. The myths in this article include the following statements:

- Myth #1: SCA is for military radios
- Myth #2: SCA is too large
- Myth #3: SCA is too slow
- Myth #4: CORBA is too slow
- Myth #5: SCA is too expensive
- Myth #6: SCA version 4.1 is CORBA agnostic

If we consistently consider all these provisions, then, as it seems to us, the authors of [8] were somewhat hasty with rebuttals.

SCA is for military radios. Yes, this is another software architecture. It is yet another component-based architecture. But did it eventually find some use outside of SDR? Component architecture depends on the availability of components. That is, ultimately, from supporting the developers. Can you provide such examples for SCA? The paper says, for example, that SCA can be used in the automotive industry. But where and by whom is it used?

SCA is too large. Here we can agree. Nordiasoft leads a good performance. NordiaSoft customers have platforms that require less than 32 megs of RAM for a complete system. And that includes: a real time operating system kernel, a TCP/IP networking layer, a flash file system, a POSIX layer, the drivers and SCA platform components for the speakers, the microphone, the transceiver, the GPP, the DSP, the FPGA, the NordiaSoft SCA Core Framework, the ORBexpress RT CORBA stack from OIS, and an SCA application made of 3 components.

SCA is too slow. Here, the authors' arguments refer to the fact that JTRS (see above) was used on the old hardware architecture. But this is a comparison of SCA with the performance of SCA in the past. It would be correct to compare the performance of SCA and other solutions on a modern hardware base.

CORBA is too slow. The authors rightly argue that this applies specifically to TCP / IP, which is the default for all ORBs. But this does not change the essence of the matter. Transmission over TCP/IP is part of the CORBA architecture and the overhead becomes very significant. We note in particular that, for example, Integrated Modular Avionics (IMA) - and this is the built-in real-time software on board the aircraft, does not use TCP/IP. Rather, we need to talk about the possibility of using an alternative transport layer with CORBA (and this is possible). CORBA is rarely specified for new systems. But, yes, there are plenty of live systems from the early 1990s, when CORBA was popular.

SCA is too expensive. Let us cite the paper [8]: “Radio manufacturers have published measurements that indicate using the SCA significantly reduces the porting costs and time-to-market”. So, they only confirm “myth” 1: there was no market for SCA outside of software radio. And in software radio, this is used precisely because you need to transfer old programs. Nobody writes new applications for this platform. Note that the company NordiaSoft, which supported SCA and whose employees the work [8] was written, no longer exists.

SCA version 4.1 is CORBA agnostic. In reality, the history of SCA has confirmed the fact that the adoption and success of software tools are determined by one simple conclusion: whether they allow you to create software faster or not. And nothing more. Everything else is some external requirements (restrictions) that must be observed. But the real success for developers is precisely the reduction of time-to-market. In this regard, we can mention another product, also, by the way, based on CORBA - Parlay. It also “failed” for exactly the same reasons. He complicated the development instead of simplifying it. But - like any product that has taken place and been used to one degree or another, SCA has not disappeared without a trace. For example, the OMG Space Telecommunication Interface (STI) specification [11] explicitly says the following: “The predecessor was a “lightened” framework based on/inspired by Software Radio and SCA Specification and followed many of the same architectural patterns." That is, the ideas of SCA will be in demand in more modern systems.

III. On Strategic Communications

A. *DISN cybersecurity – an unsolved issue due to software*

According to SSA Single Security Architecture (SSA) [12], Joint regional security stacks (JRSS) are the main components of the Joint Information Environment (JIE) environment providing a unified approach to the structure of cybersecurity as well as protecting computers and networks in all military organizations (Fig. 7).

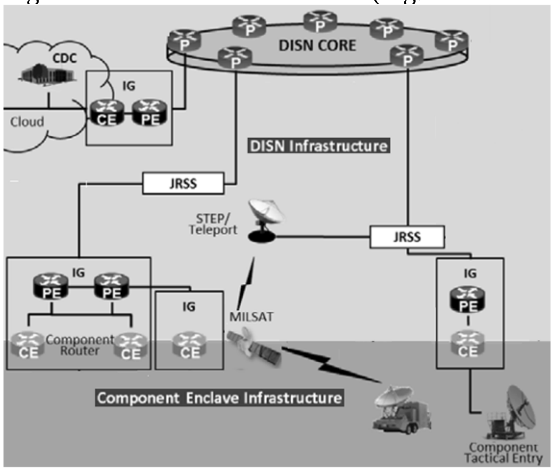

Fig. 7: The leading role of JRSS in DISN Infrastructure [13]

JRSS is a suite of equipment that performs firewall functions, intrusion detection and prevention, enterprise management, virtual routing, and forwarding, and provides a lot of other network security capabilities. JRSS equipment, in fact, is an IP-router with a complex set of cyber-protection software. For example, the typical physical NIPR JRSS stack is comprised of as many as 20 racks. The total amount of work includes the installation of 23 JRSS stacks on the NIPRNet service network and 25 JRSS stacks on the secret SIPRNet network.

But… a report GAO-16-593 [14] required more control over the Nevertheless Chief spending of funds for JRSS. And in 2018, under the pressure of GAO critics, the Pentagon’s chief weapons tester said the DoD should stop deploying its new network security platform JRSS [15]. Thus, the Pentagon suspended the $2 billion cybersecurity project.

Why? The main reason was the complexity of the software. Particularly, it characterizes the set of requirements for potential JRSS developers: knowledge of at least two or more products from ArcSight, TippingPoint, Sourcefire, Argus, Bro, Fidelis XPS, Niksun FPCAP, Lancope, NetCool, InfoVista, and Riverbed. Note that each of these companies provides its own complex software for cyber defense. How to combine them? How to hire such high-level software developers to work in a top-secret environment? The crucial JRSS failure is extremely important: JRSS is too S-L-O-W [16]. This does not satisfy the basic requirement of communication systems - their speed.

The Defense Department's newly released cloud strategy - could it be more successful? A key technological difficulty for the JEDI project is the interoperability of clouds, and once more it relates to software [17].

B. *Defense Red Switch Network*

In conditions of cyberwar, no reason to be surprised that the Defense Red Switch Network (DRSN) uses 40 years old ISDN technology. DRSN is a dedicated telephone network, which provides global secure communication services for the command and control structure of the United States Armed Forces and the NATO Allies (Fig. 8). The network has been maintained by DISA and has secured for communications up to the level of Top Secret.

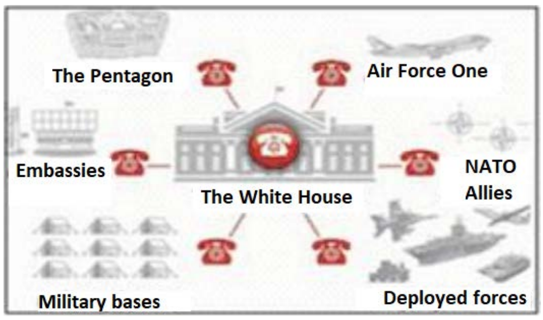

Fig. 8: The DRSN architecture [18]

Secure Terminal Equipment (STE) is the U.S. government's encrypted telephone communications system for wired communications. STE is designed to use ISDN telephone lines which offer higher speeds of up to 128 kbit/s and are all digital, can also be utilized for data and fax transmission through a built-in RS-232 port. STE sets look like ordinary high-end office desk telephones. All cryptographic algorithms are in the crypto card (Fig. 9).

Fig. 9: STE desk set. Note the slot in the front for Crypto PC Card

The Secure Communications Interoperability Protocol (SCIP) is a US standard for secure voice and data communication, for circuit-switched one-to-one connections (including GSM), not packet-switched networks (!). SCIP supports a number of different modes, including national and multinational modes which employ different cryptography. There are several components to the SCIP standard: key management, voice compression, encryption, signaling plan for voice, and data and multimedia applications [19]. It is similar to a dial-up modem in that once a connection is made, two SCIP phones first negotiate the parameters they need and then communicate in the best way possible. For voice, SCIP simply sends a stream of voice data frames (typically MELPe frames) sending a 54-bit data frame every 22.5 milliseconds. A synchronization block is sent roughly twice a second in place of a data frame [20].

C. DISN infrastructure

The target DISN infrastructure (IP-based) contains two level switching nodes: Tier0 and Tier1 (Fig. 10). Top level Tier0 geographic cluster typically consists of at least three Tier0 SoftSwitches.

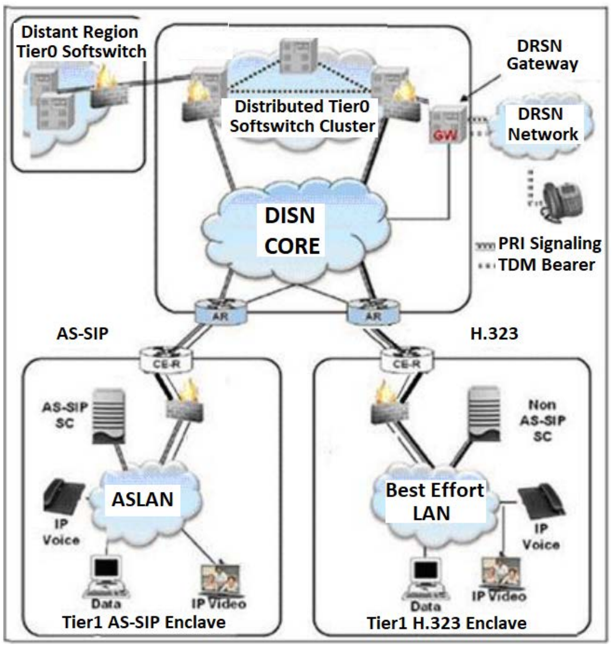

Fig. 10: The target DISN Classified VoIP and Video Signaling Design [21]

As the distance between the clustered SoftSwitches must be planned so that the return transmission time does not exceed 40 ms and propagation delay equals 6 μs/km thus the distance between Tier0 should not exceed 6,600 km. The classified signaling environment uses a mix of protocols: the existing vendor-based H.323 and AS-SIP signaling. The use of H.323 has allowed during the transition period to all DISN CVVoIP (Classified VoIP and Video). In addition, a unique MG capability exists as part of a Tier0 SS. Important note: Classified VVoIP interfaces to the TDM DRSN via a proprietary PRI and DSRN gateway (for transfer from ISDN to IP signaling and much more).

Summing up. It is still difficult to predict the time during which the DISN network will finally switch to the AS-SIP protocol. Obviously, TDM and ISDN equipment could stay for an unpredictable time, especially considering cybersecurity threats.

D. On unsolved mobile DRSN

On SCIP packet loss. Let us note that SCIP is for circuit-switched one-to-one connections (including GSM) only, not for packet-switched networks (!). In [22], the measurements of SCIP encrypted voice transmission over low-quality packet radio links are carried on. The performance of SCIP operating in an asynchronous communication network with various levels of packet loss was investigated and found inadequate mainly due to problems with cryptographic synchronization between the transmitting - receiving units and other reasons that are critical for deployed forces, at least (Fig. 8):

- Signaling state depends on the start message, multipoint cryptosync frames, and end message. If any of them is lost, the whole superframe will be discarded by the receiving unit.
- If the start message is lost, the receiving unit will remain in the receive signaling state until the time is out.
- As the voice traffic states depend on the transmitting and receiving units to be synchronized, packet losses are problematic.

On SCIP packet blocking. Note the IP network layer does not implement SCIP as a protocol since SCIP operates at the application layer in endpoints. However, the IP network layer should enable SCIP traffic to transparently pass through the network. Some network devices do not recognize SCIP, and thus remove the SCIP codecs from the SDP media payload declaration. When the SCIP media subtype is removed from the SDP media payload declaration, SCIP endpoint devices will not operate on the network. The purpose of the recent IETF draft [23] is to provide enough information to enable SCIP payloads to be transported through the network without modification or filtering. It discusses that encrypted audio and video codec payloads are transported over RTP. In any case, the fate of the target DISN design (Fig. 10) remains in question. The use of SCIP in packet radio networks is unclear till now.

IV. Conclusion

The paper considers some secure mobile communication software issues from both the tactical and strategic points of view and some failures due to overly complex software as well as cyber threats. These miscalculations or more precisely the gross failures concern Future Combat Systems (FCS), the largest and most ambitious program in U.S. Army history, and the Joint Tactical Radio System (JTRS), supposed to replace about 30 different military radio systems. A key failure is seen in the CORBA programming methodology.

What about Strategic communications, they, first, relate to the Defense Information System Network (DISN) cybersecurity. Joint regional security stacks (JRSS) are the main components of the cybersecurity structure to protect computers and networks in all military organizations. Unfortunately, in 2018, the Pentagon suspended this $2 billion cybersecurity project due to unsolved software issues. In conditions of cyberwar, no reason to be surprised that the Defense Red Switch Network (DRSN) for top secret communications uses 40 years old ISDN technology. The Secure Communications Interoperability Protocol (SCIP) is developed for circuit-switched one-to-one connections (including GSM), not packet-switched networks. How to develop the mobile DRSN version is unclear now - due to SCIP packet loss and SCIP packet blocking. This is one hard task.

Some myths of CORBA and Software Communications Architecture (SCA) are discussed and some research areas are pointed out in this context. In conditions of cyberwar and the need for top secure telecommunications, the very transition from circuit switching to internet technologies and packet switching seems doubtful.

REFERENCES

[1] Sneps-Sneppe, M., Namiot, D., and Tikhonov, E. "On Software Defined Radio Issues," *2022 Workshop on Microwave Theory and Techniques in Wireless Communications (MTTW)*, Riga, Latvia, 2022, pp. 35-40, doi: 10.1109/MTTW56973.2022.9942482.

[2] Bard, J., and Kovarik Jr., V.J. Software defined radio: the software communications architecture. John Wiley & Sons, 2007.

[3] GAO, Defense Acquisitions. Key Decisions to Be Made on Future Combat System. GAO-07-376 (Washington, DC: Mar. 15, 2007).

[4] Pernin Ch.G. et al. *Lessons from the Army's Future Combat Systems Program*, RAND, 2012. Lessons from the Army's Future Combat Systems Program (rand.org)/ Retrieved Jun 3, 2023.

[5] Gallagher, S. "Military's 15-year quest for the perfect radio is a blueprint for failing big." Ars Technica. 6/19/2012. Retrieved Jun 3, 2023.

[6] Gallagher, S. "The Army's costly quest for the perfect radio continues", Ars Technica. 3/8/2018. Retrieved Jun 3, 2023.

[7] U.S. Department of Defense. Summary of the 2018 Department of Defense Artificial Intelligence Strategy: Harnessing AI to Advance Our Security and Prosperity, Feb 12, 2019.

[8] Bernier, St. et all. The Enduring Myths of the Software Communications Architecture. NordiaSoft. White Paper 2018.

[9] *Networking handheld radio*. AN/PRC-154 Rifleman Radio. Thales Group. Retrieved Jun 3, 2023.

[10] Akeela, R., Dezfouli, B. "Software-defined Radios: Architecture, state-of-the-art, and challenges." In Computer Communications. Vol. 128, Sept 2018, 106-125. https://doi.org/10.1016/j.comcom.2018.07.012

[11] Handler, L., and Briones, J. "OMG's Space Telecommunication Interface (STI) Submission Discussion." OMG Technical Meeting. Object Management Group (OMG), 2021.

[12] Metz, D. "Joint Information Environment Single Security Architecture (JIE SSA)," DISA, 12 May, 2014. http://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/80 1001p.pdf?ver=2018-09-10-082254-477/. Retrieved Jun 3, 2023.

[13] DoD Instruction 8010.01. Department of Defense Information Network (DODIN) Transport. September 10, 2018. http://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/8010 01p.pdf?ver=2018-09-10-082254-477/ Retrieved Jun 3, 2023.

[14] GAO-16-593, "Joint Information Environment: DOD Needs to Strengthen Governance and Management," Jul 14, 2016

[15] Pomerleau, M. "The Pentagon is moving away from the Joint Regional Security Stacks," Nov 1, 2021. https://www.c4isrnet.com/it-networks/2021/11/01/the-pentagon-is-moving-away-from-the-joint-regional-security-stacks/ Retrieved Jun 3, 2023.

[16] Williams, L.C. "Is it time to rethink JRSS?" Feb 01, 2019. https://defensesystems.com/articles/2019/02/01/jrss-pause-report-williams.aspx/ Retrieved Jun 3, 2023.

[17] Keelan, T. "The Pentagon's JEDI cloud strategy is ambitious, but can it work?" March 21 2019. https://www.c4isrnet.com/opinion/2019/03/21/the-pentagons-jedi-cloud-strategy-is-ambitious-but-can-it-work/ Retrieved Jun 3, 2023.

[18] Army Regulation 25–13. Information Management. Army Telecommunications and Unified Capabilities. Headquarters Department of the Army. Washington, DC. 11 May 2017.

[19] SCIP Signaling Plan. Revision 3.2. December 19, 2007. National Security Agency. 266 pages. https://cryptome.org/2012/07/nsa-scip.pdf Retrieved Jun 3, 2023.

[20] Polak, R. et all. Secure voice transmission in the coalitional communication, Proc. SPIE 11055, XII Conference on Reconnaissance and Electronic Warfare Systems, 110550R (27 March 2019). https://doi.org/10.1117/12.2524831 Retrieved Jun 3, 2023.

[21] US Department of Defense, "Information Enterprise Architecture Unified Capabilities. Reference Architecture," Version 1.0, January 2013.

[22] Sundin, A. Audio quality perception of SCIP encrypted voice transmission over low quality radio links. Thesis, Juli 2016. Uppsala University. ISSN: 1401-5757, UPTEC F16039.

[23] Hanson, D., Faller, M., Maver, K. RTP Payload Format for the Secure Communication Interoperability Protocol (SCIP) Codec. Internet-Draft: draft-ietf-avtcore-rtp-scip-05. Expires: 30 September 2023.

Manfred Sneps-Sneppe - Ventspils International Radio Astronomy Centre Ventspils University of Applied Sciences (email: manfreds.sneps@gmail.com)

Dmitry Namiot – Coldbeans (email:dnamiot@gmail.com)