SAND2019-5077C

Trusting Embedded Hardware and Software in Treaty Verification Systems

Jay Brotz

Sandia National Laboratories
PO Box 5800
Albuquerque, NM, USA 87112
E-mail: jay.brotz@sandia.gov

Abstract:

Treaty verification equipment in bilateral or multilateral disarmament verification treaties has unique and challenging requirements for trust. With more intrusive verification concepts utilizing a wider variety of measurements and chain of custody technologies, more complex and custom equipment will be needed. Each treaty partner must have confidence that the verification equipment is behaving as expected and the host nation, in particular, must have confidence that sensitive nuclear information is not released to the inspectors. Many tools have been and continue to be created to support such verification concepts, and the process of trusting those tools, referred to as certification and authentication, have emerged as a primary challenge. We present a conceptual framework for trusting the electronics and software that control such equipment. This paper discusses a framework for trusting these programmable logic elements by a series of inspections to gain trust throughout the development cycle.

Keywords: treaty; certification; authentication; verification

1. Introduction

Trust is a key requirement in any treaty verification activity. Verification activities have several key components: an agreement between parties, a series of measurements, analysis of the results of those measurements, and a judgement[1]. The agreement, to maintain, reduce, or otherwise restrict key assets, such as nuclear weapons, drives the verification measurements, which include the procedures and the equipment to be used. Measurements, in this usage, are any on-site data collection, which can mean anything from sophisticated and time-consuming radiation spectrum collection to an inspector viewing an object. The analysis of those measurements, leading to the judgement of the agreement, is only useful to each party if the measurements themselves are trusted. While trust is often framed as a personal willingness to accept another party's declarations, far more important is the trust that each party has in the measurements that are intended to verify their agreement[2].

In bilateral or multilateral nuclear disarmament agreements, trust in the equipment and procedures of verification measurements takes on a few dimensions that pose similarities and differences to those measurements taken for physical security or international nuclear safeguards. All three domains have a similar need for trust in equipment and personnel to provide correct data and for a verification system to provide complete data. In addition, all three domains have needs for protecting sensitive information. Disarmament treaties, however, have two (or more) parties that have different needs, unlike physical security, in which one party is protecting assets and information from a broad array of adversaries, and international nuclear safeguards, in which a central authority representing a highly multilateral treaty confirms declarations of each of its member states. In a bilateral nuclear weapons limitation treaty, the limitations have historically been (and in many cases will be in the future) reciprocal, so that each party is both limiting their own stockpile and verifying that the other party has done the same. This creates the need for on-site inspections for each party in the other's territory, during which each party plays the role of host and monitor. The host is concerned primarily with convincing the monitor of their compliance with the agreement and in protecting sensitive information that is not part of the agreement. The monitor is concerned primarily with receiving sufficient evidence – collected data that can be effectively analysed – to make a compliance judgement. In a multilateral treaty, there is the potential for a party to be monitor

without the reciprocal role of being host; however, in this case the concern of the monitor is the same as in the reciprocal situation – to gather sufficient evidence to make a compliance determination.

New verification concepts will require greater functionality in verification systems in the future, leading to greater complexity[3]. This complexity, enabled by general advances in private industry, is often embodied in the hardware and software of programmable logic elements. Sensors remain important, but software can extend the functionality of a sensor or a group of sensors beyond what has been produced in the past, and often at a lower cost than sensor improvements. A key question is how the host and monitor can trust more functional and complex verification equipment in which the complexity is allocated to programmable logic elements, such as software or firmware running on a processor[4], or a programmed logic device, such as a field programmable gate array (FPGA)[5]. Whereas trust may be gained on a simpler, less functional system by visual inspection (or other means to verify the physical construction of a piece of equipment, such as radiography), the verification of programmable elements cannot simply be visual. Trust in these elements must be proven in other ways.

2. Certification and authentication

The community of developers of treaty verification systems use the terms certification and authentication to describe how parties to an agreement gain trust in equipment[6,7]. Certification is the process by which the host gains confidence that the equipment used for treaty measurements is safe for use in their facilities and will not reveal sensitive information that is not part of the agreement. Authentication is the process by which the monitor gains confidence that the equipment used for treaty measurements is correct and complete. These processes may be similar but have different requirements. Certification involves multiple stakeholders in the host party that represent facility safety, physical security, and information security. The host is concerned about limiting the information given to the monitor to that which is necessary for compliance and no more. Authentication gives the monitor confidence that equipment is providing true, correct data, and has not been tampered to give a false impression of compliance.

The effort involve in certification and authentication processes are dependent on whether equipment is host-provided or monitor-provided. Other models are possible, such as the host providing some components and the monitor others and assembling the equipment in a joint fashion prior to use. Another model is the use of third-party or commercial-off-the-shelf (COTS) equipment, though even in this case, “host-provided” or “monitor-provided” should be defined as the last party to have sole control of the equipment. If the host procures COTS equipment and has it ready prior to an inspection by the monitor, it must be considered host-provided.

The effort by the monitor to assess *completeness* in measurements is a system-level task. The effort by the monitor to assess *correctness* and the effort by the host to protect sensitive information are equipment-level tasks. The host uses equipment certification to gain confidence that the equipment has no exploited vulnerabilities that could reveal sensitive information to the monitor. The monitor uses equipment authentication to gain confidence that the equipment is designed to give correct data and that it has not been modified to give incorrect data that would support a compliance determination. In each case, the activities can be summarized as ensuring that the equipment exhibits the agreed functionality and has no additional functionality.

3. A framework for trust in new hardware and software development

While there are cost benefits to using COTS measurement equipment, treaties may have such unique sets of requirements that no COTS equipment is suitable, and therefore new equipment must be designed and built. This section describes a framework for certifying and authenticating custom treaty equipment from the requirements through to system operation, moving through the development cycle stage by stage to ensure the major objectives of confirming that the system exhibits the agreed functionality and that it has no additional functionality.

This agreed functionality can be captured as well-defined requirements that specify the allowed functions. If these functional requirements are specific and complete, any function in a system or system design that is not described in the requirements is disallowed by the agreement. Therefore, the

requirements become the functional reference that is used to trust the system design, and then the built system. The requirements serve as the functional reference for trusting the rest of the development flow, as seen in Figure 1.

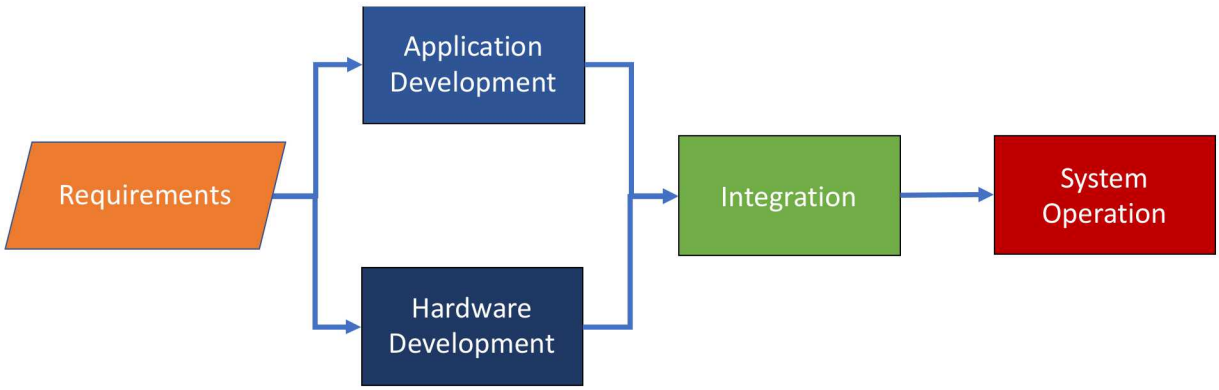

Figure 1: A generic system development flow

Once the requirements are developed and agreed to by all parties to the treaty, they can be used to authenticate or certify the system design, regardless of the party that creates the design. As seen in Figure 1, design can be divided into hardware design and application design (the outputs of the hardware development and application development processes, respectively)[8]. These design outputs can be verified by physical inspection (by visual means including radiography, thermal imaging, etc.) or digital inspection.

In each step of the development cycle, the output of the design or build process is compared to a reference to confirm the inspection objectives. That reference is the trusted output of the previous process. For example, within the Hardware Development stage in Figure 1 could be the creation of an electronic circuit on a printed circuit board (PCB). The stages of developing that PCB could include the definition of the circuit that results in a graphical schematic, the synthesis of that schematic into a netlist, and the layout of that netlist in the physical space of the PCB, as indicated in Figure 2.

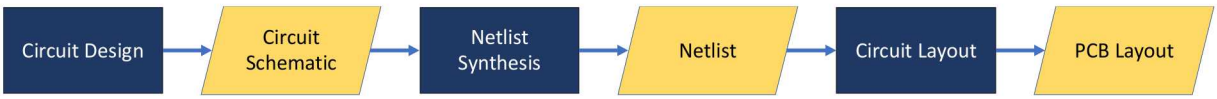

Figure 2: An example development flow for a printed circuit board

In Figure 2, the rectangles are development processes and the rhombuses are development outputs. During each inspection, a process output is compared to a reference that is the previous trusted process output in order to confirm that it still exhibits all agreed functionality with no additional functionality. As an example, the circuit schematic output of the circuit design process is compared against the requirements for this hardware component. When the circuit schematic is found to have the required functionality with no additional functionality, it becomes a trusted reference for the next inspection, in which the synthesized netlist is inspected, as shown in Figure 3.

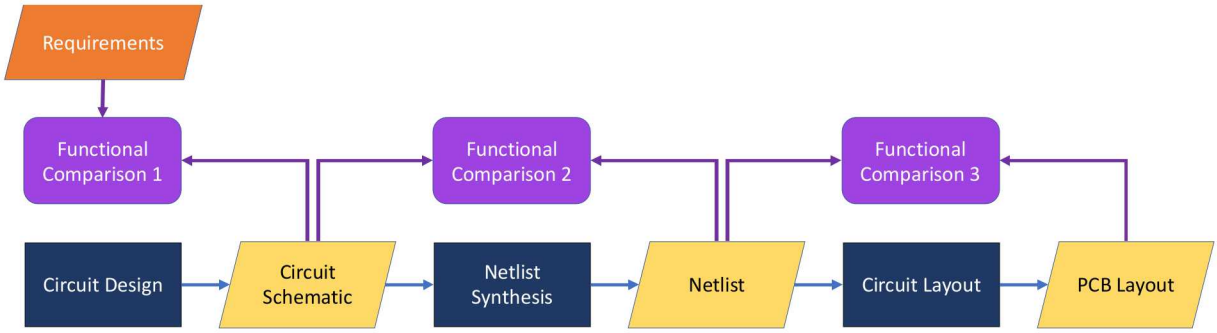

Figure 3: Functional comparisons of outputs in the example development flow of a printed circuit board

Most of these inspections against a reference work in one of two ways: reference comparisons can be exact or functional. In an exact comparison, the known output is already trusted, and the specific output is compared against it. For example, compiled software that resides on a piece of equipment can be compared to a golden copy of that compiled software (either in its entirety or by using a hash) and if any bit is not the same, it will fail. A functional comparison, on the other hand, can be used when a known, trusted output does not exist yet, or that output can vary in form depending on the specifics of the process used to create it (for example, compiled software varying by the compiler used and the compiler's settings). In a functional comparison, the output is analysed for its functionality and then compared to the reference list of functions (potentially going backward in the development flow all the way to the requirements). For example, in Figure 2, the output netlist can be functionally compared (with the right tools) to the circuit schematic, or to the requirements themselves, to understand whether it exhibits the agreed functionality and has no additional functionality. That netlist then becomes a trusted output.

The entire system development flow for a particular equipment design can be constructed as a flowchart that begins with requirements agreed to by the treaty partners and ends in system operation. It would be like the flowchart in Figure 1 with each process stage expanded to the level of detail in Figure 2. This flowchart will be a directional graph with no loops; that is, despite the number of parallel processes that occur, every process will have inputs that are closer to the requirements and outputs that are closer to the final system operation than the process itself. As such, every system development flow can be trusted by functional or exact reference comparisons starting from requirements and moving, process by process (and output by output) toward the system operation. A logical way to utilize these inspections is to use functional comparisons throughout the development process to authenticate or certify a system as it is being designed and built. A system could be designed and built entirely by one partner and presented as a complete system, which could then be authenticated or certified by functional comparison of the built system with the requirements, though the inspections involved may be more difficult to perform and have greater uncertainties in that case. Once a functional comparison has resulted in a trusted output, that output can then be used as a reference for exact comparisons. An example of this use is with compiled firmware: the firmware machine code could be compared to the trusted source code through a functional comparison, and when found to be in accordance with the agreed objectives, be used as a reference for an exact comparison of the loaded firmware the next time the system is used.

While exact comparisons can be designed to have very small uncertainties, functional comparisons are likely to leave some room for doubt when yielding a positive ("matching" the trusted reference) result, and functional comparisons will require different tools and procedures for different development flow processes. In many cases, the functional comparison tools are themselves an area of research or non-existent and must be created or adapted from other tools for a specific equipment authentication or certification. There are classes of functional comparison tools that could be used in many situations for different pieces of equipment. These include:

- Comparing software source code to requirements
- Comparing compiled software to source code
- Comparing FPGA hardware description code to requirements
- Comparing FPGA synthesized netlists to hardware description code
- Comparing FPGA bitfiles to synthesized netlists
- Comparing hardware circuit schematics to requirements
- Comparing hardware circuit designs (such as PCB layout) to schematics

- Comparing fabricated and populated PCBs to PCB layouts

4. Evaluating functional comparison methods

Functional comparisons for certification and authentication inspections are a research need, and this research would benefit from a common evaluation method. Each of these inspections compares a development output to a trusted reference output from a previous process to verify that the functionality is the same, and that there is no additional functionality in the new output. Effectiveness of that comparison can be measured with the metrics of sensitivity (true positive rate) and specificity (true negative rate).

The sensitivity of the comparison method measures the proportion of positive results (or functional matches) that were correctly identified as such. The true positive test can be conducted with a number of different designs undergoing the same development process and applying the comparison method to the outputs of that process, as shown in Figure 4. Since the output should functionally match the reference, each positive result from the comparison method would contribute to the true positive rate.

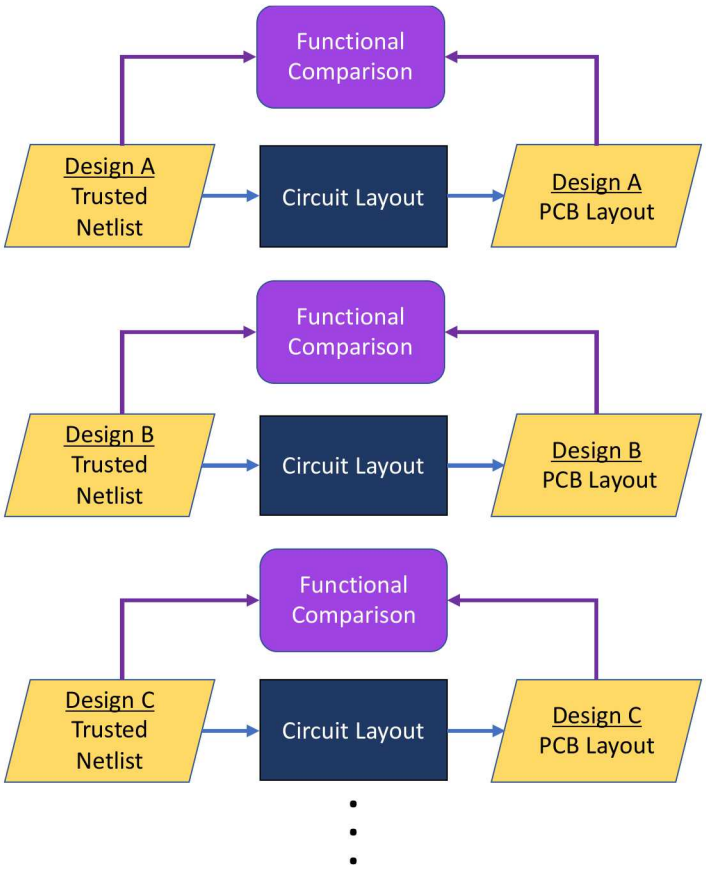

Figure 4: Sensitivity tests of a functional comparison method with three separate designs in the same development process

The specificity of the comparison method measures the proportion of the negative results (or cases with functional differences) that were correctly identified as such. The true negative rate test can be conducted with a set of modified inputs to a development process that create a functionally modified output and comparing that modified output to the trusted (unmodified) input, as shown in Figure 5. For example (seen in Figure 5), for a PCB layout process, a trusted netlist could be modified by adding or removing functions in the form of circuit branches and elements, and then used to create a layout with an autoroute tool followed by manual finishing. A functional comparison method could be evaluated by comparing a range of various modifications in the circuit layout to the trusted netlist. Since the output is

functionally different than the input in this comparison, each negative result would contribute to the true negative rate.

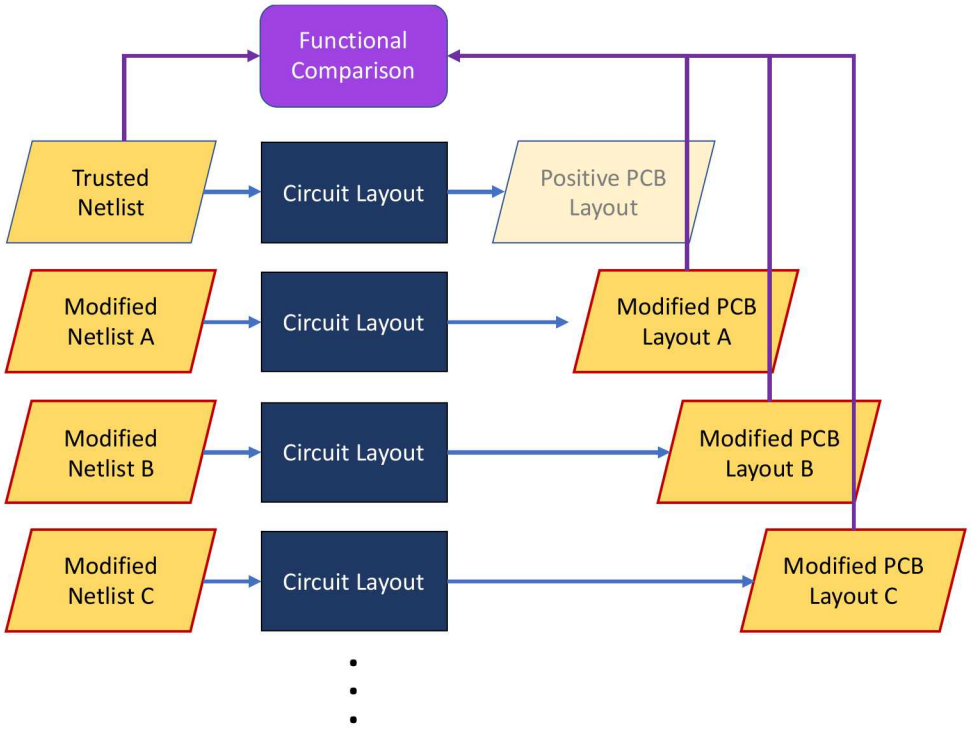

Figure 5: Specificity tests of a functional comparison method with three modifications of the same design compared to the trusted input

Maximizing the volume of data from these tests will improve the evaluation of a functional comparison method for sensitivity and specificity. In addition, the complexity of the designs and the nature of the modifications may have an impact on the ability of the functional comparison method to exhibit a high sensitivity and specificity.

5. Recommendations for a path forward

A broad array of inspection techniques could be useful for certification and authentication throughout the development cycle of custom treaty verification equipment. These techniques that perform functional comparisons, especially for design outputs for the programmable elements of a system, should continue to be developed. These inspection techniques should be developed in parallel to the development of the treaty verification tools themselves, to produce custom treaty verification equipment that is more inspectable and, in the eventual case of use in a treaty or agreement, more trustable by all parties.

The development of the verification tools and the inspection techniques are neatly separable into activities that can be performed by two different organizations in collaboration, where one organization conducts “blind” tests of the other’s development outputs. This can serve as a model for international collaboration on nuclear verification in a number of fora.

Finally, tools for functional comparisons in related industries, such as integrated circuit verification used for error prevention by chip manufacturers, or software verification tools used for high consequence software applications, should be analysed for their utility in this framework.

6. References

[1] Avenhaus, R., Kyriakopoulos, N., Richard, M., and Stein, G., (Eds.); *Verifying Treaty Compliance: Limiting Weapons of Mass Destruction and Monitoring Kyoto Protocol Provisions;* Springer-Verlag; Berlin and Heidelberg, Germany; 2006.

[2] Wuest, C. R.; *The Challenge for Arms Control Verification in the Post-New START World*; Lawrence Livermore National Laboratory, Livermore, CA; 2012.

[3] Woolf, A.; *Monitoring and Verification in Arms Control;* Congressional Research Service; 2011.

[4] Kütt, M., Göttsche, M., and Glaser, A.; *Disarmament Hacking 2.0: Toward a Trusted, Open-Hardware Computing Platform for Nuclear Warhead Verification;* Proceedings of the Institute for Nuclear Materials Management Annual Meeting, 2016.

[5] Brotz, J., et. al.; *FPGA Authentication Methods: US-UK Collaboration for Treaty Verification;* Sandia National Laboratories and AWE; Albuquerque, NM, USA and Reading, UK; 2017.

[6] MacArthur, D., Hauck, D., and Thron, J.; *Simultaneous Authentication and Certification of Arms-Control Measurement Systems;* Proceedings of the Institute for Nuclear Materials Management Annual Meeting, 2012.

[7] Tolk, K. and Benz, J.; *An Architectural Approach to Authentication and Certification of Arms Control Equipment;* Proceedings of the Institute for Nuclear Materials Management Annual Meeting, 2018.

[8] Brotz, J. and Hymel, R.; *Framework for Evaluating Authentication Methods for Treaty-Related Processor Systems;* Proceedings of the Institute for Nuclear Materials Management Annual Meeting, 2016.

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.