PNNL- 13131

The Importance of Detailed System Documentation for Measurement Systems Authentication Under Cooperative Monitoring Arrangements

Bruce D. Geelhood
James L. Fuller

January 17, 2000

Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest National Laboratory
Richland, Washington 99352

THE IMPORTANCE OF DETAILED SYSTEM DOCUMENTATION FOR MEASUREMENT SYSTEM AUTHENTICATION UNDER COOPERATIVE MONITORING ARRANGEMENTS

INTRODUCTION

Under various conceivable warhead dismantlement transparency and/or sensitive fissile material control cooperative monitoring arrangements, authentication of the monitoring equipment will be required. Typically such equipment might be used to provide confidence to monitors that the material is of weapons origin, is of weapons grade, or has some other collection of properties considered by the host to be classified sensitive information. For these cases the United States, to some degree in cooperation with Russian specialists, is working to develop measurement systems which provide the monitors with the requisite confidence, but also to protect the host country's information. This protection will be accomplished through the use of information barriers. It is the position of the United States Government that the best way to assure the protection of the host's sensitive information is for the host to supply, certify, or otherwise have the right of last private inspection and keep on its soil any measurement system used for these purposes. Under this scenario, authentication processes will need to be devised to insure the monitoring party that the host-supplied equipment is making credible measurements.

There are three fundamental approaches to authentication:

- Random selection of equipment
- Use of trusted, unclassified calibration sources
- Thorough joint inspection using detailed design documentation.

For a given monitoring situation, it is likely that specialists will use a combination of procedures from each of these approaches to devise an overall authentication process. Additionally, and it almost goes without saying, cooperation and confidence is greatly enhanced if the design of the integrated measurement plus information barrier system is performed in a completely open, cooperative, joint manner.

THE PROBLEM OF THE "HIDDEN SWITCH"

Random selection of equipment provides the monitor with a very powerful tool to select the actual piece of host-supplied equipment to be used to make a measurement, and to take away another piece to destructively examine using private means. This approach becomes less feasible for expensive equipment, sub-systems, or modules. It becomes impractical for large, installed measurement systems.

The use of trusted calibration sources is a good approach to verify that the measurement system under question can function properly. It is even conceivable, assuming that there are no operational security of health physics limitations, that through clever procedures employing host-blind source selections by a monitor, that a monitor might gain additional confidence that the installed system has not been configured for covert control by the host.

This paper specifically addresses the importance of extensive installed equipment inspection using detailed as-built design documentation as the most effective way to assure a monitor that host-supplied equipment, operated in a host-owned facility, can not be covertly controlled by the host for his benefit. We refer to this issue as the "hidden-switch" problem. The hidden switch problem is examined in the context of the weapons-origin attribute measurement demonstration planned for early in CY 2000 at Los Alamos National Laboratory for Russian technical experts associated with U.S.–Russian Fissile Material Transparency Talks (FMTT).

A "hidden-switch" denotes anything that can selectively compromise the monitoring results in a manner favorable to the host. For FMTT, the favorable result would generally be acceptance of the contents of a canister as weapon-origin material. This "hidden switch" spoof will be considered here in a generic fashion to avoid classification issues. It will also be considered in the context of the Information Barrier Design Criteria established under the auspices of the Joint DOE-DoD Information Barrier Working Group. These criteria are listed in Appendix I.

The radio-frequency shielding design basis criteria addressed this "hidden switch" concept in regard to the concern that the host must be prevented from taking over the measurement system results by remote control. The "host-supply" concept of the design basis indicated that authentication would then become more difficult.

To effectively implement a successful "hidden switch" spoof, the host must be able to selectively and reliably trigger the switch whenever necessary to erroneously pass a canister that would not properly pass the inspection criteria. That is, the host must implement a method of covertly altering or switching the results displayed to the monitoring party in response to the trigger signal.

The use of a relatively sophisticated trigger signal can be forced on the host by procedures. If the system must be demonstrated to operate properly in a "open" mode with calibration sources, the system must have hardware and software sufficient to provide both the correct result and the erroneous result. In addition, it can not use a trivial trigger that passes all items. If the system can be shown to properly fail test objects while in "classified" mode, the "classified" Mode State of the system can not be used as a simple trigger for passing all classified items.

Authentication measures must recognize that the host can attempt to hide the trigger and/or switch in either hardware or software. There are several possibilities for the host to design covert features into the measurement system. All these features require extra components or software to trigger and implement the cheat. The detailed documentation aids the monitoring party to ferret out these extraneous items given sufficient access and resources.

HARDWARE DOCUMENTATION

The system documentation can be used to search for extraneous components in either the hardware or software that the host plans to use to implement a cheat. Consider an example following the design basis requirement to block host use of radio frequency (RF) remote control. It is recommended that a RF enclosure be used both to prevent remote control and to protect the classified information. To implement RF remote control, the host requires an extraneous RF receiver module to trigger the desired cheat. The documented effectiveness of the attenuation of the RF enclosure impacts the frequency selection, the size and location of an antenna, and the sensitivity of the receiver. Detailed examination of the documentation may reveal receiver components (e.g., antenna, a high-Q tuning coil, a demodulator, etc.) or signal paths extraneous to system functionality.

The hardware documentation for authentication purposes requires a detail level sufficient to allow the monitoring party scientists and engineers to ferret out potential "hidden switch" features. Clearly, this includes overview documentation sufficient to understand the measurement techniques and to reproduce the measurement parameters. However, the level of detail required to ferret out a potential "hidden switch" feature far exceeds that necessary to provide a general system design understanding.

The detailed electronic-component documentation should include descriptions, block diagrams, schematics, and photographs of all the electronics. It would be to the monitoring party and honest host's advantage if the monitoring party could carefully examine these before any in-person inspection. The electronics documentation should include the following:

- Block Diagrams showing all logical functions
- Descriptive narratives explaining each function with references to the component parts and identifiers used
- Schematics identifying component parts and showing all interconnects
- Printed-circuit (PC) masks for each PC board layer showing all interconnects
- PC board loading diagrams identifying component parts by identifier and value (e.g., R123 & 100 ohms).
- Photographs of both sides of the finished and loaded PC boards
- X-rays of components that can not be readily opened for examination (e.g., sensor heads, multilayer boards, sealed power supply blocks).

This level of documentation would allow the monitoring-party engineers to reverse engineer the as-built system sufficiently to discover extraneous components or interconnections possibly associated with receiving a trigger to cheat or switching data or results to accomplish the cheat. This level of documentation would also allow the inspector to be briefed on where to look for covert or undocumented components or connections.

The monitoring-party would be aided by negotiated constraints on electronic designs to enhance inspectability. For example, single-sided or double-sided PC boards are more inspectable than multilayer boards. If multilayer boards are used, the internal layers should be constrained to power and ground connections only. Thorough authentication of multilayer boards

requires X-rays or extensive continuity measurements in association with the masks if any undocumented interconnections are to be found. Inspectability is also aided if components are mounted with identification marks up. Interconnection paths are more easily verified on PC boards than on wire wrap designs, which require exhaustive continuity checks. Presumably, wire-wrapped circuit boards and modified printed-circuit boards would not be acceptable components for an IB system due to inspectability issues.

The monitoring-party would also be aided by negotiating that all the computers within the IB system be identical single-board computers. Note that use of different computer types or models results in different microprocessors, BIOS firmware, support chip sets, board layouts, documentation, and possibly operating systems. If several identical computers are utilized, only one set of computer hardware documentation must be examined in detail greatly simplifying the authentication task. If only one computer is allowed in the entire IB system, the hardware-to-documentation comparison task is also simplified. The use of only one computer model or only one computer is technically achievable.

Prior inspection of the documentation allows inspectors to obtain more information on specific components from the web or requested vendor documentation. This would allow inspectors to recognize dual-use components. Any sophisticated cheating scheme will likely require extra memory or processing capacity than normally required. Thus, under-utilized components should be suspect.

A joint inspection of the entire IB system can be accomplished before each use. The thoroughness of the inspection will depend on the purpose and the level of detail in the documentation used for the comparison. A "good-faith" functionality inspection compared to a pilot's checklist could be accomplished in under an hour. The checklist would prevent inadvertent human errors by insuring that the system was properly connected and external adjustments properly made. A detailed inspection intended to ferret out a "hidden switch" spoof might take all day provided the monitoring party has done their homework with the previously provided detailed documentation. The time and preparation required for a spoof inspection can be reduced by taking clear and detailed "as-built" pictures.

The joint inspection could be conducted by a three-person team. A host technician would be allowed to touch the system to open for visual access and to make any requested measurements. A scientist or engineer from both the host and monitor teams would be allowed to view the components and to select measurement test points.

The joint inspection would allow careful visual comparison to photographs and radiographs provided in the detailed documentation. This would establish that each PC board etching and component loading was fully documented, acceptable, and unchanged. Careful examination of connectors and the wires leading from the bulkhead mounts might reveal extraneous trigger paths for cheating. This visual inspection could include taking detailed photographs of the equipment for the monitoring party to further inspect at home.

The joint inspection should perhaps allow field X-rays to be made of components that can not be opened (e.g., HPGe sensors, neutron tubes, CPU chips, power supply blocks, and multi-layer boards). Field X-rays could prevent hidden modifications or exchanges.

The joint inspection team could randomly request verification of test points against the detailed documentation. A battery-operated DVM, oscilloscope, and logic analyzer could be used by the host technician. Illustrative oscillographs should be included in the documentation showing typical amplitudes, widths, shapes, and polarity of analog signals at several test points. These signals should be checked with and without calibration sources in place, especially if the functionality of the system is questionable due to a suspected malfunction. The multiple monitored signals and complex triggering of a logic analyzer would be necessary to view the transfer of digital computer information and handshaking protocols. Some nuclear pre-amplifier designs provide input power on the signal output line since the high-frequency signal and DC power are easily separated within the pre-amplifier. Viewing signals along each interconnection (oscilloscope or logic analyzer trace) would aid ferreting out covert dual-use features.

The joint hardware-inspection/functionality-testing could include benchmarking the computers against the documentation to insure the CPUs are functional. For example, speed and accuracy test programs should produce results consistent with CPU specifications or previously run values. The PROM containing the benchmarking programs would most likely be different than the PROM used to operate the IB system to reduce the amount of software subject to line-by-line inspection. This joint inspection should also view the setup values of the computer to make sure the values exactly match documentation.

This joint visual inspection should especially search for any extraneous device that the host could use to trigger a cheat or the monitor could use to extract information.

An on-site joint visual inspection has limitations. It can not find covert modifications within hermetically sealed components without X-ray capabilities. Exhaustive functionality testing would likely be excessive, but selective functionality testing would enhance confidence. Visual inspections cannot inspect software or firmware, which the next section addresses. However, if the monitor has in his possession an complete set of system documentation as described above, then he has the major advantage of randomly selecting a particular module, subsystem, or even component for inspection or perhaps even change-out, putting the host at much more risk in trying to hide a "switch."

SOFTWARE DOCUMENTATION

The software documentation is fundamentally different from the hardware documentation. A copy of the software itself can be inexpensively provided as part of the documentation package. Commented source code can be self-documenting to a large degree. If each line of code is explained by an adjacent comment, the monitoring party is greatly aided in any line-by-line examination. The monitoring party can not physically see software components or exhaustively monitor operations in-situ during a visual inspection of the hardware. Several factors make thorough software inspection more resource consuming than hardware inspection. The

software comes in two forms human readable (commented source code) and machine readable (executable code). The software code can execute much more varied and complex operations than the non-CPU hardware in typical measurement systems. It is not as easy to examine software outputs for every input combination as for hardware components. The software code could conceivably be self-modifying in a manner not available to hardware.

The software is a very attractive location for a "hidden switch" that alters the results. It is much easier to modify a few data values in a HPGe spectrum than to alter fast electronic pulses with high accuracy. The software code necessary to easily implement a raw data swap or modification can be very simple and small relative to the code within the computer. The software code necessary to implement discarding actual results and substituting desired results can be exceedingly small and difficult to locate. In some cases, changing only a single computer instruction might be sufficient to favorably alter results. Only line-by-line examination of the source code would be capable of finding data or result swaps because there are potentially a large number of places to place this spoof. Software also has the ability to process complex decision trees to determine when to trigger a cheat.

The Russians historically have been suspicious of American computers and software. Thus, extra opportunity should be allowed to verify and authenticate the software. Although providing an identical copy of the system hardware may be expensive, duplicate copies of the software can be very inexpensive to provide.

The operating system, data collection, and data analysis software will reside on a programmable read-only memory (PROM) chip. It would be a measure of the host's good faith to prove that two identical PROM chips contain the same code and then allow the monitors to randomly select one for use and the other to take home for private examination as part of the software documentation package. There are at least two methods for proving PROM chips identical.

One method would require a "monitor-supplied" comparison computer that would read the contents from the two PROM chips and confirm that they are identical. An alternative to "monitor-supplied" is to allow the monitor to retain ownership after the comparison. This computer would require a read-only I/O card designed to read PROM chips and some simple comparison software. The software would sequentially read the entire memory from each PROM into a large disk file and then compare the two files. It could write that file to disk for the monitors to avoid risk of losing the contents due to electrostatic damage while carrying the PROM home. An alternate might be to jointly program the PROM from a CD-ROM or disk file that is subsequently provided to the monitoring party.

The other method would use some hash function to compare the contents of the two PROMs. This might be easily accomplished if two identical single-board computers with identical PROMs are presented for random selection. Then in "open" mode with keyboards and displays attached, the hash function programs could be run with several monitor selected base values entered and the hash functions recorded by pencil and noted to be the same for both computers. This requires writing the hash-function comparison code and writing a report explaining that this is an adequate comparison. The hash function must 1) be totally transparent, 2)

test every byte of PROM memory and 3) prevent intentional compensating modifications. In this case, the PROM and the computer would be carried home in one package.

With either method, the monitoring party should be provided the commented source code for all software within the PROM. Compiled code should be provided with a copy of the commercial compiler used and all compiler parameter settings. Thus, after a line-by-line examination of the source code, the compiled *.EXE or *.COM files could be byte-for-byte compared with those in the PROM. For commercial operating system and compiler files, the monitor would have the option of comparing the executable files with copies independently procured. The monitor would have the option of a line-by-line examination of operating system code.

Commented source code on disk is a fundamental aspect of any detailed software documentation package. The necessity of line-by-line examination has been discussed previously. The purpose of the software documentation package (including the source code) is to allow the monitoring party the ability to ferret out any "hidden switch" feature and to aid that search. A potential "hidden switch" feature might be found in the provided source code or by finding a discrepancy between the code used and that generated by the provided source code. Tracking the software logic and the "hidden switch" search would be aided if the software documentation includes the following:

- Scientific papers should describe analysis and algorithms.
- Flow charts or block diagrams should clearly describe the flow of the program logic.
- Documentation should include descriptions of what each routine or subroutine inputs and outputs along with an overview of the operations it performs. The description should include the purpose and the methods used. The description should also include a list of calling programs and subroutine/functions called.
- Source code should explicitly define all variable names/types/definitions with comments describing the variable and relating it to symbols used in papers describing analysis and algorithms.
- Source code comments should adequately explain each programming step.
- Notes should call attention to and justify software features that might aid covert functions. Such features include 1) self-modifying code, 2) indirect methods to access data (FORTRAN EQUIVALENCE, C UNION, and POINTERS), and/or 3) interrupt driven routines.

The line-by-line inspection of all the system software will be a huge and expensive task. The monitoring-party would be aided by negotiated constraints on software designs. For example, the choice of operating system has a huge impact on both the ability to authenticate and the resources required. Clearly, WINDOWS would be much more difficult to authenticate than DOS. The monitoring party will be aided by requiring that all computers in the IB system use identical operating systems with all extraneous features removed. The monitoring party will also be aided by prohibiting self-modifying code.

Authentication of analysis software can be aided by electronic search tools if both the commented source code and the executable code are provided on disk. This allows monitoring-party programmers to use editor search functions to track variables through the code. This also

allows a companion search in the executable code or an intermediate assembly language listing for references to variable addresses. The design basis alluded to direct I/O references, which allow the executable code to be electronically searched for I/O instructions and port addresses. The software validation can also be aided by running the code under debugger control with watch points set to indicate when selected variables are altered. The debugger should be able to locate self-modifying code if watch points note any unexpected alterations in code segments. It is possible to run software under logic analyzer control to trigger on execution at a specific address, execution of a specific operation, detection of an interrupt request, or operation on a specific variable. The logic analyzer can then disassemble surrounding machine code into "as-executed" assembly language listings. This feature could be used to track interrupt calls or self-modifying code.

Validation of analysis software can be aided by file output of intermediate results. For example, validation of a peak-fitting program benefits from being able to compare peak center, width, and amplitude values with those from a program that the monitoring party trusts. If the only output from the PU600 analysis program is the plutonium isotope ratio, the monitoring party should have the option of viewing intermediate results to gain confidence in the answers. This may require adding output lines to the provided source code and running it on test data files. The validation team might also desire to run pathologic cases through the analysis software to determine the robustness of the software as a means of reducing failures to pass due to system errors.

CONCLUSIONS

A detailed design documentation package is essential to the transparency of an information-barrier-protected system. It serves the monitoring party to find any "hidden-switch" spoof. It provides the monitoring party with the ability to randomly select a subsystem, module, or even component for very close examination, thereby putting the host at much more risk to trying to implant a hidden switch. Any gap in the documentation suggests a potential location for a "hidden-switch" spoof. This paper illustrates what is required in a documentation package and how it might be used to authenticate an IB system. The paper points out some electronic tools that may aid in the authentication task if the correct documentation is provided in a useful format. Lack of host cooperation regarding documentation, could be indicative of something to hide.

APPENDIX 1

Radiation Detection Information Barriers Design Criteria*

- Equipment Certification

  Host country "certifies" equipment as meeting its own security requirements.

- Central Processing Unit (CPU)

  Use "trusted" processors (processors which are dedicated to specific tasks, having extraneous functionality eliminated, such as single board computers).

- Non-CPU Equipment

  HPGe systems and related subsystems are probably inherently inspectable. Other radiation detection subsystems must be considered on a case-by-case basis.

- Procedural Issues

  Case-by-case evaluation required, but in general instruments should be able to accommodate all anticipated variations in measurement conditions without revealing classified information.

- Electronic Emanation Considerations

  Equipment should be evaluated for emanations according to standards and practices acceptable to host.

  The monitoring party will have to perform system-level assessment or risk, and might demand rigorous emanation protection even under trusted processor arrangements.

- Multiple/Intermediate Barriers

  If intermediate barriers can be employed without compromising functionality assurances, then it may be desirable to do so.

- Software, Firmware, and CPU Operating Systems

  The software at every level must be completely inspectable and documented; amount of code must be minimized, and complex operating systems and compilers avoided.

- Inputs and Outputs

* As formulated by U.S. Joint DOE-DoDWorking Group on Information Barriers, 1999.

All I/O must have a well-understood, dedicated function, with no extraneous ports/devices associated with the integrated inspection system; simple displays should be used for yes/no type output results, peripherals must be minimized, and bus structures avoided.

- Measurement System Authentication and Repair

  Multiple copies of host-provided equipment should be maintained under secure storage, with monitoring party selecting one for inspection upon demand. Software should be similarly supplied on demand, particularly prior to first use. Most defective equipment should be discarded and replaced, and detector head checks an exception.