LA-UR- 09-06225

Approved for public release;
distribution is unlimited.

Title:	Designing Minimum Functionality Attribute Measurement Hardware
Author(s):	Peter J. Karpius Richard B. Williams
Intended for:	Submission to NA-241 as deliverable for FNI-029 task.

Los Alamos
NATIONAL LABORATORY
EST. 1943

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Form 836 (7/06)

Designing Minimum-Functionality Attribute Measurement Hardware

P.J. Karpius and R.B. Williams

ABSTRACT

Previous efforts to produce attribute measurement instrumentation for treaty verification and nuclear transparency initiatives have focused on demonstrating that the measurements can be automated and that any sensitive information concerning the item under test can be protected to the satisfaction of the host entity. For the most part, these systems have only approached authentication as an afterthought, and have run into serious real-world feasibility problems as a result.

In this paper, we present a series of design criteria for creating attribute measurement hardware for which authentication is one of the primary drivers, on par with protection of the host entity's sensitive information. The assumption being that in order for an attribute measurement system to be of any practical use, both the host and inspecting parties must believe that the output from the attribute measurement is acceptable.

We will discuss impediments to authentication which affect the selection of hardware components, and the results these selections may have on output accuracy. We will also cover the core functionality requirements of a neutron/gamma attribute measurement system designed for measuring plutonium items and the minimum hardware capable of carrying out these tasks. Finally, a discussion of the compromises between ease-of-authentication and flexibility of operation is given.

Designing Minimum-Functionality Attribute Measurement Hardware

P.J. Karpius and R.B. Williams

ABSTRACT

Previous efforts to produce attribute measurement instrumentation for treaty verification and nuclear transparency initiatives have focused on demonstrating that the measurements can be automated and that any sensitive information concerning the item under test can be protected to the satisfaction of the host entity. For the most part, these systems have only approached authentication as an afterthought, and have run into serious real-world feasibility problems as a result.

In this paper, we present a series of design criteria for creating attribute measurement hardware for which authentication is one of the primary drivers, on par with protection of the host entity's sensitive information. The assumption being that in order for an attribute measurement system to be of any practical use, both the host and inspecting parties must believe that the output from the attribute measurement is acceptable.

We will discuss impediments to authentication which affect the selection of hardware components, and the results these selections may have on output accuracy. We will also cover the core functionality requirements of a neutron/gamma attribute measurement system designed for measuring plutonium items and the minimum hardware capable of carrying out these tasks. Finally, a discussion of the compromises between ease-of-authentication and flexibility of operation is given.

INTRODUCTION

Systems designed to carry out nuclear transparency initiatives, such as verifying nuclear warhead dismantlement, must satisfy the requirements of all entities party to a governing treaty or agreement. The requirements of the host, whose material or items are to be inspected, are often in conflict with the needs of the inspecting party. That is, nuclear material of weapons origin may possess traits that the host deems to be sensitive and, for reasons of their own national security, they are unwilling to disclose such information to any inspecting party. Such traits may include the material's mass, geometry, or exact isotopic composition. The host must be convinced that the verification system is certified not to release such sensitive information during verification.

The canonical solution to this problem is to design a device that performs possibly sensitive measurements on an item but only releases minimal output in a format that has been determined not to contain sensitive information by the host entity. The design of such a piece of equipment would necessarily be a cooperative effort between the host and inspecting parties. Long before the instrument was built or used, both groups would have to agree that, as designed, the output would be simultaneously non-sensitive and relevant for inspection purposes.

One active research path towards this goal has been that of an "attribute measurement system" (AMS). The AMS technique begins with detailed automated measurements of nuclear material using standard non-destructive assay (NDA) techniques. These detailed measurements are then used to derive properties of the material such as plutonium mass or age. These values are then compared against some pre-defined thresholds and one or more pass/fail results are given as output. Because the output is so information-sparse, there is little danger from the host's perspective of an

inadvertent release of sensitive information—at least via the approved output channel. The challenge is to choose the measurements and thresholds such that the very limited output is useful to the inspector. This presents a challenging diplomatic problem to be addressed during negotiation of the governing agreement—how are the thresholds themselves selected without giving away sensitive information? The solution to this problem is highly scenario-dependant and is outside the scope of this paper.

For the time being, let us assume that diplomatic efforts have succeeded in a bilateral agreement that an AMS technique is to be used, a set of NDA measurements to be undertaken by the AMS has been identified, and the thresholds for comparison have been selected. The problem is then given to the technical teams from both sides to design the AMS that can make these measurements, undertake the threshold comparisons, and provide the output in a manner that is secure, verifiable, and authenticatable. Previous AMS demonstration systems such as the FMTTD and NG-AMS have shown with high confidence that we can automate radiological measurements, calculate relevant derived quantities for meaningful attributes, and perform threshold measurements all while keeping the instrument operators and observers separated from any sensitive internal data by means of an information barrier. The crux of the issue is now to design a system that can make the same measurements and calculations as the previous iterations of AMS hardware, but is also authenticatable.

MEANS OF AUTHENTICATION

Exactly how one would go about "authenticating" an instrument is not something that is well-defined. Much would depend on the context in which the instrument was being used. Is there a credible threat that the host might maliciously tamper with the device in order to alter the output? Is authentication just being used as a means to ensure that benign errors in manufacturing or handling of the equipment have not been made? Is the inspector allowed unfettered access to the hardware immediately before and/or after the measurements are taken? Is there a trusted chain of custody between the inspection site and a location where invasive, destructive testing can be performed? What constitutes sufficient confidence that the device is built as designed and has not been altered?

These are issues which must be resolved ahead of the design process as part of the negotiation between the host and inspecting parties. In lieu of an actual measurement campaign and negotiation, we must now either conjure a hypothetical scenario towards which to build or try to find characteristics that are likely to be common amongst a variety of likely measurement scenarios.

We posit that in the realm of nuclear weapons dismantlement treaty verification, both parties involved are well financed, technologically adept, and have a very strong stake in the outcome of the inspection. For these reasons, we assume for the time being that there is a credible threat for malicious behavioral modification of the system by the host. Furthermore, we posit that the host entity will have sole control of the hardware for some non-negligible period of time prior to the start of measurements, during which such behavioral alterations could be put into effect. If this is the case, then the opportunity for inspector authentication falls into three regimes:

1. Pre-measurement authentication
2. Authentication during measurement
3. Post-measurement authentication

Pre-measurement authentication includes inspection of the instrument, visually and/or with the use of other non-destructive analysis techniques (e.g. eddy current scanners, query/response tests of software, photogrammetric analysis, etc.). The techniques must obviously be non-destructive, but the implementation of the technique must also pass muster with the host—the inspector must not be seen as having the opportunity to alter the behavior of the device such that sensitive information is released. We believe it is likely that any pre-measurement authentication that is allowed to take place will be cursory and limited.

Authentication during measurement is similarly hampered, this time by the design of the instrument itself—the information barrier, which keeps the sensitive information secure, also prevents much in the way of diagnostic data from reaching the inspector. Gross malfunction can be detected, but detecting inappropriate yet well-formed responses requires some clever testing of the state space. If diplomatically possible, an initial portion of the measurement campaign should involve measuring a series of reference sources that test the threshold space of the AMS. If the inspector is allowed to determine which reference source *or sources* are in the measurement container without the host knowing, then you can go a long way in ruling out the possibility of real-time control of the output state by the host entity. Unfortunately, there are still ways around this test. Specifically, the device could be altered to give the wrong answer only when a specific isotope is present—one that is known not to be among the reference standards, for instance. While host-blind testing is an important step for building confidence, it should not be viewed as fool-proof. Without seeing what the device is actually composed of, we cannot say for certain that the behavior of the device will be what we want it to be.

Post-measurement authentication must be relied upon for the lion's share of inspector confidence building. At this stage in the process, there is no longer a strict need for the instrument to be functional, so destructive techniques become a possibility. There is a significant security question involving the inspector being allowed to touch/measure/take equipment that was involved in the measurement of the host's sensitive nuclear material. While it seems unlikely, sufficient diplomatic pressure from a high enough level can work miracles in this sort of scenario. It is possible that the inspectors could have access to some or all of the measurement components once the material had been removed. Failing that, a system of random selection implemented at the beginning of the measurement campaign could allow for the inspector duplicates of some or all of the instrument modules. The confidence that the modules are indeed duplicates is a function of how effectively the random selection is implemented, how many clones there were to choose from and how many of the spares the inspector is allowed to analyze. The specifics of random selection are outside the scope of this paper.

KEEPING THE SYSTEM MODULAR AND/OR COMPACT

For the time being, let us assume that the inspector has in their possession at the conclusion of the measurement campaign one or more instrument modules, which they believe are either the actual units used in the measurement or at least identical copies. There may be a certain amount of analysis and testing that can be performed on site, but this is a highly non-optimal analysis scenario. Any authentication analyses that take place at the host facility run the risk of being monitored. If the host is aware of what techniques are used to authenticate the device, they have the upper-hand in devising a work-around behavioral modification. If at all possible, the in-depth authentication

analysis work should be performed at a trusted location with no host presence whatsoever. For this, we need to assume a means of transport of the devices to be tested from the measurement location "back home." This is something that becomes increasingly involved the larger and more mechanically complex the device is.

As far as chain of custody is concerned, the smaller and lighter the better. Ideally, the entire device would be something that could be placed in a backpack and carried off site by the inspectors themselves. This presents difficulties when trying to implement an entire gamma/neutron NDA measurement system and should be viewed more as a guideline for design rather than a goal. Making key functional components (as distinct from structural components, etc.) modular and removable aids in both chain of custody as well as mix-and-match style random selection.

- *Compact and/or modularized hardware facilitates authentication.*

MINIMIZING STATE SPACE

If we are able to get an instrument out of the measurement campaign that we believe is identical to that used in the measurement itself, we have a lot of options for testing its behavior. The first test would be to plug it in and turn it on, run it through some test measurements and ensure that the device responds correctly to various stimuli. This process becomes considerably more meaningful if the AMS has been designed in such a way that its entire state space can be explored without having to use inputs or stimuli that aren't part of the standard operation of the instrument.

The state diagram for a simple electronic circuit may only have a handful of states and one or two paths leaving from any one state. A desktop computer, on the other hand, has as many states as there are combinations of CPU register values and memory contents—effectively trillions. The wealth of instructions that can be executed at any cycle, combined with the fact that programs are able to modify executable code on the fly, makes the idea of writing a true state diagram for the entire computer essentially impossible. Trillions of states with countless inputs interconnecting the states in a quagmire of complexity—this is not a system that can be fully tested operationally. You might make a program whose *internal* state diagram is quite simple, but it is difficult to prevent an outside process in the computer from modifying the program memory and therefore the state diagram, often without the knowledge of the modified program. If we want to perform meaningful operational testing, we must have hardware and software that are simple such that the number of states and interconnections between states can be controlled.

- *Simple hardware and software facilitates authentication.*

MINIMIZING THIRD-PARTY SOFTWARE TOOLS

Most software, even for embedded applications with simple processors, is written in high-level languages and then compiled into machine code that the processor can understand. The reason for this is that high-level languages work much like our spoken languages and allow us to think about what the code is doing in a natural way. Generally, machine code just looks like gibberish

and without knowing the key for decoding it—that is to say, *de*compiling the code back into some sort of pseudo high-level language—we cannot understand that at which we are looking. This might seem like a great argument for writing all software that requires authentication in a high-level language such as Java or C.

Unfortunately, there are hidden steps involved in going from a high-level language like C to machine code that is useful to the processor. In particular, the code must be "compiled" which requires the use of a very complicated tool known as a compiler. The code may then also be combined with external libraries—precompiled bits of machine code that perform common functions such as floating point math routines or input/output handling—using another tool known as a linker. Depending on the high-level language chosen, and the toolset used to transform it into machine code, there may be several third-party tools that touch the code and have the opportunity to affect its operation. All of this is transparent to the user.

There have been demonstrated compromises of this third-party interaction wherein perfectly valid code with no security holes is compiled into an executable containing security holes via a malicious compiler. Thus, even if we have the source code for the software or firmware that our instrument is running, and have decided that the source code is authenticated, *and* have verified that when the valid code is compiled we get the same binary machine code that was used in the actual measurement, we still can't say that the code that ran during the measurements was valid. We must also authenticate all software that touched the source code during its transformation into machine code.

To make matters worse, the various compilers, linkers, etc. available to use are also written in high-level languages and were compiled and linked with who-knows-what. At some point with high-level languages, we have to draw the line and say that we've gone back far enough through iterations of compiled compilers that we believe intervention is unlikely. But before you write this off as paranoia, consider Ken Thompson's "Trusting Trust" (*Communication of the ACM*, Vol. 27, No. 8, August 1984, pp. 761-763). Mr. Thompson was able to write a compiler that recognized the source code for the Unix login() command and inserted a backdoor password that always worked. Thus, whenever his tainted compiler was used to compiled valid login() source code, a tainted login() executable would be created. Furthermore, the compiler would also recognize when it was being asked to compile a new version of itself—and inserted the malicious code into the new compiler as well! Any descendants of that initial compiler would always generate faulty login() binaries, even if the user compiled login with a compiler which they created themselves from source code they trusted. This sort of attack is not to be dismissed. Third party software tools may be difficult to stay away from in some circumstances, but authenticating them is not a closed-loop operation.

- *Third party software tools hinder authentication*

MINIMIZING CODE SIZE

Even if third-party tools are used, there is always a direct means of checking that malicious code has not been inserted into your binaries. Unfortunately, it's arduous: you must go through the binary file instruction by instruction and recreate the source. Essentially, decompile the code by

hand. Of course there are tools that will do this automatically for you, but using one simply reintroduces the third-party software problem.

Beyond being simply tedious, manual de-compilation and reconstruction of source is fantastically difficult. As software complexity rises, the time required to accomplish this task goes up exponentially. Furthermore, if the same source code was compiled by two different compilers or on two different architectures, the output binary will be different, often drastically. A compiled binary that has been manually vetted, when compared against another compiled binary from the same source made some time later when the compiler had been updated to a new version might not match at all.

Software and firmware that is written directly in assembly language is the least prone to these problems; there are no third-party tools required and, because coding in machine language is difficult and slow, software developers tend to put a lot of effort into minimizing the number of instructions required to perform a given task. Short code makes de-compilation feasible and also reduces the possible complexity of operations. Reduced complexity leads to decreased opportunity for malicious code to be hidden somewhere inside.

- *Smaller code base facilitates authentication*

MINIMIZE HARDWARE COMPLEXITY

Along the same lines as minimizing code footprint, the process of authenticating the hardware portion of the instrument also scales with complexity. Just as we must look at the raw code line by line in order to see what is truly happening with the software, we must look at the individual hardware components piece by piece to discover the true functionality of the hardware. This includes dissecting integrated circuits and determining the internal mapping of transistors, resistors, and capacitors to reconstruct a schematic which is the hardware equivalent of source code. Failing to do this in-depth analysis invites the introduction of "sneak circuits" or alterations to existing circuitry to modify behavior.

Performing an in-depth hardware reverse engineering (RE) is a laborious process and must be done one component at a time. Thus, simple reduction in the number of components will linearly affect the time and expense required for the RE process. There are two ways to approach this reduction in complexity. The first is to reduce the complexity of what is being performed such that less hardware is needed to accomplish the goal. The second is to avoid using any products that incorporate functionality that will not be used in the normal functioning of the device.

- *Reduced hardware complexity facilitates authentication*

MINIMIZE COMMERCIAL SYSTEMS

The NG-AMS project relied heavily on commercial off-the-shelf hardware whenever possible. The presence of unused hardware (and software) components became a real problem when it came to authentication. The trouble with commercial systems is that they are typically designed to be flexible and useful in as many disparate environments as possible, thus increasing

the profitability of the product. Unfortunately, the goals of profit and authentication are essentially completely at odds.

Commercial products tend to incorporate hardware and software components that are unnecessary for the needs of the AMS, though these components must still be authenticated in order to ensure that there is no hidden functionality there that might affect the output of the AMS. This adds to the authentication workload, wasting time and resources on components that do not forward the goals of the instrument. Additionally, commercial products tend to be ensconced in proprietary technologies, closed-source software and firmware, intellectual property concerns, etc., which do not make authenticating what they've done more straightforward.

- *Commercial systems impede authentication*

FAVOR HARDWARE OVER SOFTWARE

It has been established that every component of hardware and every instruction of software must be authenticated to achieve the highest confidence that the operational configuration of the instrument is understood. Unfortunately, both of these tasks are time consuming and difficult. However, there is a distinct difference between hardware reverse engineering and manual software decompiling—there is a large, active, and mature industry surrounding hardware reverse engineering. The same procedures necessary to confirm the schematic of an integrated circuit are used every day in the fields of intellectual property law and technical competitive analysis. There are numerous companies, both domestic and international, that perform these services with high throughput at reasonable costs. The government also maintains this capability for its own internal vulnerability assessment needs.

Manual software decompiling, on the other hand, is essentially never done except by computer science undergraduates with cruel professors. The one exception to this is in the field of computer virus response, in which a piece of malicious code is decompiled to discover how it works. The throughput here is very slow and the code segments tend to be incredibly short. For an excellent example, see "The Internet Worm Program: An Analysis," Purdue Technical Report CSD-TR-823, by Eugene H. Spafford. The capability exists to perform the needed analysis, but the industry to do so efficiently does not. In terms of time and available resources required, software reverse engineering is considerably more expensive than hardware reverse engineering.

- *Hardware reverse engineering is more cost effective than software reverse engineering*

ONE POSSIBLE APPROACH

Trying to tie together all of the authentication drivers listed above has led us down a path towards an AMS design that, while perhaps not "easy to authenticate," is at least "easier" than previous incarnations. In the interest of maximizing the gain from hardware and software reverse engineering, we have attempted to use as little of both as possible—with emphasis on reduction of software. The goal is a single-board AMS that includes the analog MCA front end for the gamma

system, multiplicity shift register for the neutron system, and a simple microprocessor for calculating derived quantities and doing threshold comparisons.

For now, the neutron detectors (presumably ^{3}He tubes with poly moderator), HPGe gamma ray detectors, and associated preamplifiers are assumed to be separate hardware modules from the AMS board and are outside the scope of this paper. The following sections will discuss functional components of the proposed AMS board in greater detail.

NEUTRON DATA ACQUISITION

The specific goal of the neutron NDA measurement is to determine the value for $^{240}Pu_{eff}$, which when combined with the gamma-ray isotopic measurement can determine the plutonium's mass. The quantity $^{240}Pu_{eff}$ is the amount of ^{240}Pu that would give the same coincidence response as all the even isotopes in the item[i].

$$^{240}Pu_{eff} = 2.52^{238}Pu + ^{240}Pu + 1.68^{242}Pu$$

In standard neutron coincidence counting measurements, the value of $^{240}Pu_{eff}$ is one of the three unknowns, the other two being (1) the fission multiplication and (2) the factor α, which relates the (α, n) neutron rate to that of spontaneous fission. These three unknowns can be obtained by directly solving the three point-model equations[ii] related to the single, double, and triple neutron coincidence rates from an item. Thus, the neutron measurement hardware must keep a tally of singles, doubles, and triples over a specified measurement interval. To keep such a tally, we implement a simple shift-register circuit. The basic components of a shift-register circuit consist of a series of clock-driven flip-flops linked together in stages.[i] The circuit stores an incoming pulse train for a predetermined time so that (1) each pulse can be compared with every other pulse within that time window and (2) true coincidences can be statistically distinguished from accidental coincidences. This method enables dead time-free operation up to input count rates of several hundred kilohertz. For the proposed measurement of plutonium, count rates are expected to be quite low and dead time is not expected to be a significant factor.

We assume that the neutron detector, including its ^{3}He-tube preamplifiers, will be a COTS item or functionally equivalent to modern COTS preamplifiers such as the PDT110. Because these amplifiers already have digital signal output, the entire neutron signal processing front-end on the AMS board can be implemented in digital electronics—simple logic gates can be used to construct the entire shift register, if desired. Using 7400-series logic integrated circuits (e.g. an array of 7470 J-K flip-flop chips) would allow for the entire shift register to be built without the need for software of any kind. The contents of the shift register are read out each time a new pulse enters and these data are used to populate a histogram of multiplicity values. This can be accomplished using a series of adders and counters, again achievable using nothing but TTL logic chips.

At the end of the measurement interval, the contents of the multiplicity histogram are analyzed to determine the rate of singles, doubles, and triples—the first three factorial moments of the count distribution (see "Application guide to neutron multiplicity counting," Los Alamos Report LA-13422-M, Ensslin et al., section 5). This step is somewhat computationally complex but can still be achieved in pure logic. An example of how to construct such a circuit is given in "Development of a Portable Neutron-Multiplicity Counter and Metrological Controls", Konyaev et al., Atomic Energy, Vol. 77, No. 6, 1994. The output buffers of this circuit hold the numerical

value for S, D, and T—the singles, doubles, and triples counts that are used as inputs to the point model equations, allowing us to calculate α, M, and $^{240}Pu_{eff}$, the latter of which is the key value for the mass calculation being performed in these measurements.

Solving the point model equations is a computationally expensive process involving significant floating-point mathematics. These floating-point operations require either special software/firmware routines in order to emulate floating-point mathematics on a fixed-point processor, or a microprocessor with a dedicated floating-point engine. In short, handling floating-point operations requires an increase in either hardware or software (or both) over strictly fixed-point operations. In the quest to keep software and hardware complexity to a minimum, it is better to avoid floating-point operations altogether. However, if it is impossible to do away with them, an increase in the complexity of hardware is preferable to including floating-point software libraries in an otherwise fixed-point architecture.

In the case of the neutron multiplicity point model equations, given in Ensslin, et al., and clarified in "A note on the multiplicity expressions in nuclear safeguards," Pazsit, et al., NIMPR A 603 (2009) 541-544, it is difficult to remove the need for floating-point mathematics. Our goal is to determine whether measured values for S, D, and T lead to a value of $^{240}Pu_{eff}$ that is within some pre-determined error thresholds. Solving the point model inversion in the traditional manner absolutely requires floating-point operations, particularly a number of division operations of non-integer values.

We attempted to back-solve the point model inversions, starting with an acceptable range of $^{240}Pu_{eff}$ values and ending with an expression for the valid ranges of values for S, D, and T. In doing so, we would be able to pre-calculate metrics for judging the values of S, D, and T directly and would not need to worry about the point model equations at all during an AMS measurement campaign. Unfortunately, due to the non-linear nature of the point model equations, the back-solved results were no less complex than the equations they replaced.

In the end, we concluded that the simplest approach was to solve the point model equation inversions as described in Ensslin, et al., and as implemented in existing coincidence counting codes such as INCC. Unfortunately, this does involve the inclusion of floating-point capability. As discussed previously, we choose to incorporate this functionality via hardware rather than software. The functionality of the neutron multiplicity portion of the AMS is now reasonably defined.

- TTL signals from the neutron preamps are processed by an unintelligent shift register consisting of basic logic ICs. The result is three memory registers containing the values for S, D, and T.
- A small piece of optimized firmware solves the point model equations in order to calculate $^{240}Pu_{eff}$.
- The value for $^{240}Pu_{eff}$ is compared by firmware against hard-coded thresholds to determine whether this portion of the measurement was successful. The output of this routine is a single bit, pass or fail.

The first of these three steps is conducted without the use of software at all. For prototyping purposes, this portion can be built into an FPGA to allow for configuration changes without hardware changes. The second two steps are all software, but are reasonably simple in their capabilities. No operating system or other third-party tools would be required to achieve this functionality.

GAMMA-RAY DATA ACQUISITION

A specific goal of the gamma-ray measurement is to determine the isotopic fraction of $^{240}Pu/^{239}Pu$. This fraction serves as an indicator of a plutonium item's usefulness in a nuclear weapon.[1] A microprocessor will be used to conduct the analysis of the gamma-ray data that yields this isotopic fraction. A digitized plutonium spectrum serves as input to that microprocessor from the gamma-ray measurement subsystem. We assume that a HPGe crystal, cooling system, and preamplifier will all be COTS components. A multichannel analyzer (MCA) that forms the gamma subsystem of the minimum functionality system must discriminate, amplify, filter, and digitize these pulses.

Previous efforts have yielded successful results when using simple custom MCAs, but these devices were designed for use with spectral-template-comparison systems that recorded low-resolution NaI(Tl) data.[iii] When designing an MCA for HPGe spectra, care must be taken not to sacrifice resolution—through peak broadening caused by electronics instability—for the sake of simplicity. Although such broadening is tolerable in NaI-based systems, as a result of inferior resolution compared to HPGe, broadening could nullify the main advantage of the latter. One potential tradeoff is sophisticated data acquisition hardware, which would relax the requirement on the analysis algorithm, versus simple data acquisition hardware, which would require a more advanced analysis algorithm. Discrimination and digitization tasks are relatively straightforward and do not generally introduce resolution-hampering effects.

Preamplifier pulses are less than desirable for creating a spectrum because their long decay times cause the occurrence of pulse-pile up at even modest count rates. These long decay times also cause poor signal-to-noise in the waveform region that is far from the leading edge of the pulse in time. To optimize processing, a filter circuit is used to shape the preamplifier pulses. This shaping, which in its most basic form consists of a CR-RC differentiator-integrator, may constitute a source of resolution degradation. The differentiation stage produces a pulse with a sharply decaying tail. This tail may not return to the voltage baseline in the desired manner and may lead to subsequent tailing on the low- or high-energy side of the peaks in the resulting spectrum.

Countering this effect is a pole-zero circuit, which is essentially a variable resistor in parallel with the input capacitor of the main filter circuit. Although generally correctable, this is one example of how additional complexity is required to address the impairment of resolution by an MCA design that is too simple. MCAs intended for use with HPGe detectors should be reviewed because such sources of degradation are numerous.

Figure 1

~~Figure 1~~ shows the MCA-166 from GBS-Elektronik GmBH,[iv] a compact and popular MCA used for many years by the International Atomic Energy Agency. This MCA has reasonable resolution parameters of 610±20 eV at 122 keV for a 500-mm^2 planar HPGe detector at < 10 000 cps. The MCA-166 employs an 88C166-5M microcontroller but is operated externally via RS232 to PC-based data-acquisition software. The form factor of this MCA is small, with physical dimensions of 155 × 9.5 × 50 mm and a total mass of 1 kg.

[1] The gamma-ray subsystem also can be used to determine the date of chemical separation of americium from plutonium, but that will be an analogous task, in terms of system design, to the measurement of the $^{240}Pu/^{239}Pu$ ratio.

This device has an acceptable level of performance for attribute measurements, possesses limited complexity compared to state-of-the-art MCAs, and fits within a small package. However, its reliance on an external computer with sophisticated software does not make its design as a whole suitable for our needs.

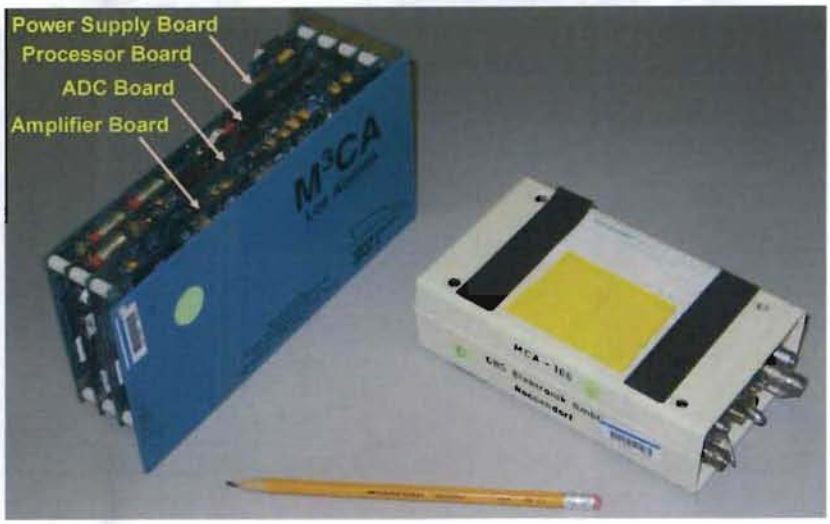

Figure 1. Two small analog MCAs: The M³CA on the left and the MCA-166 on the right.

The general issue of including more features—and hence complexity—in hardware leads us to look into MCA designs of the more distant past. In the early 1990s, a team based at Los Alamos National Laboratory developed the first portable MCA known as the M³CA[v] (Figure 1), which possessed dimensions of 10 × 20 × 9 cm. This four-board system consisted of an amplifier, analog-to-digital converter (ADC), power-supply, and processor board. The amplifier board provided two selectable time constants, fine- and coarse-gain adjustment, pile-up rejection, dead-time correction, and active-baseline restoration. The ADC was of a Wilkinson type with 512- and 4096-channel conversion gains. The power-supply board provided low voltage and bias voltage, for standard detectors of that time. The processor board was equipped with three serial ports, 16-bit binary I/O, EEPROM for parameter storage, and FLASH ROM for program storage.

As dated and simple as the technology of the M³CA is, it still has more functionality than is required for the current project. However, because the M³CA represents the lower limit on acceptable performance for this project,[vi] we can capitalize on this simple yet sufficient system by taking from its design only what is required and stripping away what is unnecessary. In general, we would remove any ability to configure the hardware and then hardwire the required settings. For example, it has been previously shown that gamma-ray attribute measurements of plutonium can be made with a few select regions of the spectrum[vii]. For instance, all that is required to obtain the isotopic ratio of $^{240}Pu/^{239}Pu$ is the region roughly encompassing 630 to 670 keV.

Because we are not performing general nuclide identification, we can expect specific peaks to be present in our measured plutonium spectrum. We then can hardwire the MCA to work only in this region, which would be reasonably covered by 512 channels at a fixed total gain but still leave room to address potential peak-drift issues. Shaping times and discriminator levels could also be fixed. We could also remove the entire pileup rejection circuit, depending on the maximum count rates to which the system could be exposed.

Firmware in the M^{3}CA can handle some spectral analysis functions, such as region-of-interest (ROI) operations and centroid calculations. These functions would not be implemented within the gamma-ray measurement subsystem but instead within the same microprocessor intended to conduct the neutron analysis.

GAMMA-RAY ANALYSIS

The ratio of the quantity of two radionuclides can be expressed in terms of the intensities of their gamma-ray peaks, half-lives, branching ratios for the selected peaks, and the relative efficiency at the energy of those peaks. For example, the following relation expresses the ratio of ^{240}Pu to ^{239}Pu based on their respective 642- and 646-keV gamma rays:

$$R = \frac{^{240}Pu}{^{239}Pu} = \frac{^{240}I(642)}{^{239}I(646)} \cdot \frac{^{240}T_{1/2}}{^{239}T_{1/2}} \cdot \frac{^{239}BR(646)}{^{240}BR(642)} \cdot \frac{RE(646)}{RE(642)}$$

In the above expression, we assume the following: (1) relative efficiencies are virtually equal at this energy separation, (2) the branching ratios and half-lives are hard-coded, and (3) the system must only determine the net peak areas and hence the intensities. By applying a simple ROI method to obtain peak areas, we can greatly simplify the hardware requirements compared to what would be needed if we pursued a least-squares fitting algorithm. As an initial step, we created a short spectroscopic analysis routine to determine processor requirements for the gamma-measurement analysis. Currently, the code employs C++, but it does not rely on object-oriented processes, and could easily be mapped to C. Figure 2 shows the ROIs, as well as their linear background functions.

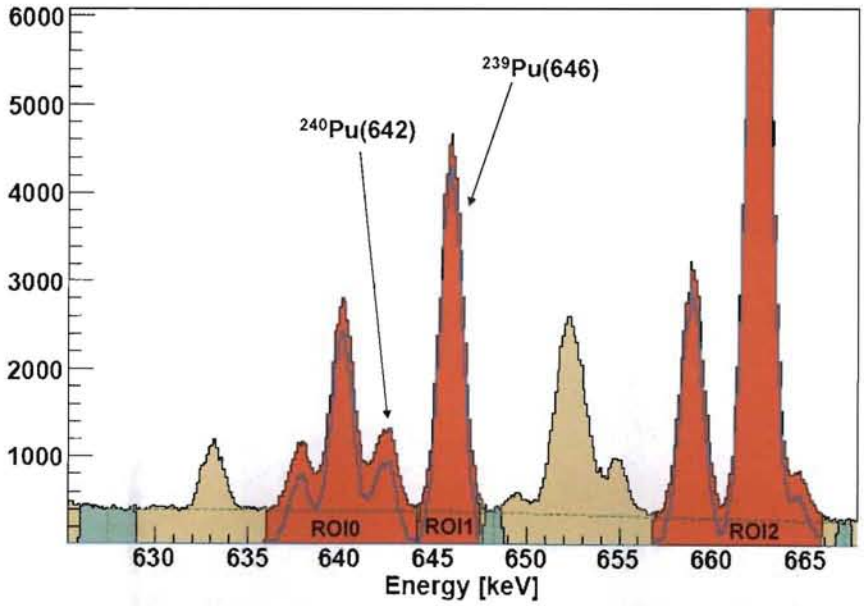

Figure 2. ROIs around 650 keV in a Pu spectrum. The three ROIs (red) represent the main regions for the analysis. The dark-green ROIs define the background regions. The green dashed lines represent the calculated linear background functions and the cyan histogram is the background-subtracted spectrum.

To better determine the centroid of the 662-keV peak of ^{241}Am, which is used for energy calibration, we first use an empirical function to smooth the raw spectrum. Using a hard-coded default energy calibration, we then determine within a specified range of the smoothed spectrum the channel with the maximum counts. This is the initial estimate of the centroid, which we use for energy calibration.

The counts in the preliminary centroid channel, as well as the two adjacent channels, are used as three values to solve for the three unknowns in a quadratic description of the peak. To yield a more accurate determination of the centroid, we set the derivative of this quadratic equation with respect to channel equal to zero. The ratio of the selected calibration energy to that of the newly determined centroid channel gives the correct energy calibration.

The code then loops over the spectrum's full range of channels and determines whether each channel lies within the ROI boundaries as determined by the energy calibration on the ROI limits. The code does address the possibility that, at an ROI's boundary, only a fraction of a channel may be contained within that ROI. The value assigned to an array element for any ROI channel consists of the content of that channel's raw spectrum scaled by the fraction of the channel that lies within the ROI's boundary. We assign the full-bin content of the corresponding spectrum channel to channels that are fully contained within the ROI.

Two background ROIs per main ROI are filled to estimate the background in the analysis region. Once the main and background ROI arrays are populated, we determine the average background in each of the latter. We interpolate a linear background function under the main ROI by using the average bin content and the location of the central channel of each background ROI. This background is then subtracted from the main ROI array channel by channel and a net ROI array is populated. We also calculate for each ROI the total gross, background, and net areas. The net area will be used to analyze isotopic ratios, whereas the gross and background areas will be used to propagate the error on the ratio.

Once we obtain the net peak areas (as described above), we determine the isotopic ratio ^{240}Pu/^{239}Pu using the 650-keV region of plutonium. Arrays are populated with the half-lives, peak energies, and branching ratios of the three isotopes in the analysis region: ^{239}Pu, ^{240}Pu, and ^{241}Am. The 646-keV peak of ^{239}Pu is the only peak that is clearly isolated; thus, we can determine its area without stripping away any other peaks. The peak's area is simply the net area of the second ROI.

The ^{241}Am peak at 662 keV is overlapped by two weaker gamma rays from ^{239}Pu at 659 and 665 keV. Using the net area of the cleanly isolated 646-keV peak from ^{239}Pu, along with the ratio of the branching ratios of the 659- and 665-keV peaks relative to the branching ratio of the 646-keV peak, we strip these two peaks from the ROI to yield the net area of the 662-keV peak of ^{241}Am. A similar treatment is applied to the first ROI to strip away overlapping ^{239}Pu and ^{241}Am gamma-ray peaks to reveal the net area of the ^{240}Pu line at 642 keV.

Note that the analysis does not use the ^{239}Pu and ^{241}Am peaks near 652 keV because there is the potential for a neutron capture line from Cd in that region.

To determine the activities for both ^{239}Pu and ^{240}Pu, we first divide the net area of the 646- and 642-keV peaks, respectively, by their corresponding branching ratios and then multiply by the half-life of the corresponding isotope. We then calculate the ^{240}Pu/^{239}Pu.

Efforts to remove floating-point operations entirely from the code, which could have simplified the hardware, were unsuccessful for reasons similar to those discussed in the neutron multiplicity section, above. Such efforts failed primarily because of factors such as the branching ratios of the gamma rays involved in the analysis. These ratios cannot be readily converted to integers while preserving the accuracy of the calculation. The fact that the neutron analysis already

requires some modicum of floating-point capability renders this problem moot. The microprocessor will already have a sufficient floating-point engine for handling the needs of the gamma analysis routines.

We tested this code in a limited fashion on both high- and low-burn-up plutonium spectra with both acceptable and poor statistics. Figure 3 shows these results, which are comparable with the advanced isotopic analysis algorithm FRAM[viii], for which the standard parameter sets for both shielded and unshielded coaxial HPGe detectors were used.

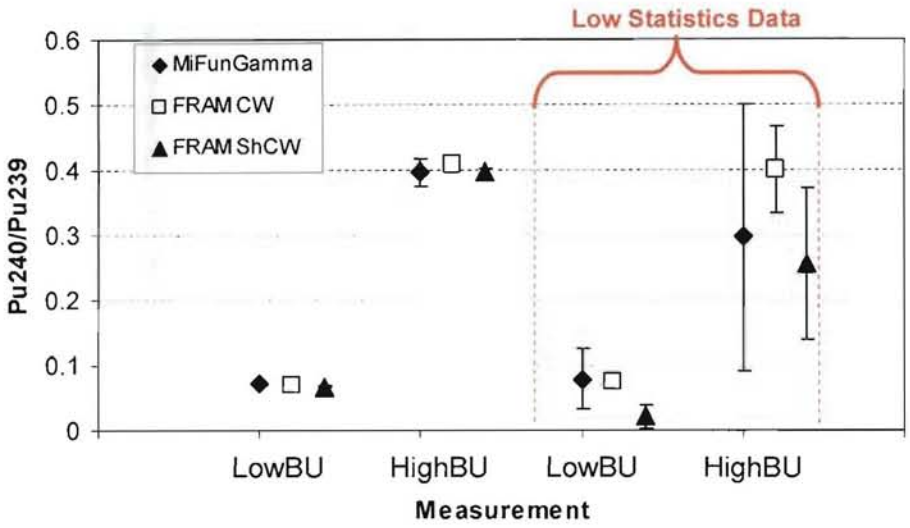

Figure 3. Comparison of Minimum Functionality code (MiFunGamma) and FRAM performance. The isotopic ratio of $^{240}Pu/^{239}Pu$ is plotted with absolute error bars for four different data sets.

COMMENTS ON COMPUTATIONAL HARDWARE

Because it is more complex than the neutron analysis, the gamma-ray analysis sets the microprocessor's level of sophistication. A processor word size of 32-bits would accommodate the above-described floating-point operations with sufficient accuracy. We adopt this word size also in the interest of standardization for authentication, by conforming to IEEE 754-2008, which requires a 32-bit word size for single-precision floating-point values.

Volatile memory would store the neutron data, as well as the gamma-ray spectrum. An authentication tradeoff between the physical complexity of SRAM and the operational complexity of DRAM warrants further investigation. Unlike the NG-AMS, the results of the attribute measurement, reduced to non-sensitive form, need not be communicated via RS232 to an external data barrier. The single microprocessor onboard the AMS can absorb the state space monitoring and output driver functions handled by the NG-AMS's data barrier, further reducing equipment footprint and part count.

CONCLUSION

We have taken initial steps in designing the neutron and gamma-ray components of an AMS based on custom hardware. Minimizing features that invariably accompany commercial components will reduce the effort required to authenticate the system. The design's level of sophistication is ultimately driven by the gamma-side minimum measurement requirements. By creating a system that is as simple as allowed by these requirements, we hope to maximize the "authenticatability" of the design and raise the level of inspector confidence in future potential realizations of this design.

---

[i] D. Reilly, N. Ensslin, H. Smith Jr., and S. Kreiner, *Passive Non-Destructive Assay of Nuclear Materials (NUREG CR-5550)*, Office of Nuclear Regulatory Research; Nuclear Regulatory Commission: Washington, DC (1991).

[ii] N. Ensslin et al., "Application Guide to Neutron Multiplicity Counting," Los Alamos National Laboratory report LA-13422-M (1998).

[iii] K.D Seager et al., "Trusted Radiation Identification System," Proc. INMM 42nd Annual Meeting, Indian Wells, CA, (2001).

[iv] GBS-Elektronik GmbH, Bautzner Landstraße 22 01454 Großerkmannsdorf

[v] J.K. Halbig et al., "Advances in and Uses of Gamma-Ray Field Instrumentation at Los Alamos," Los Alamos National Laboratory document LA-UR-94-0276 (1994).

[vi] D.T. Vo, "Extended Evaluations of the Commercial Spectrometer Systems for Safeguards Applications," Los Alamos National Laboratory report LA-13604-MS (1999).

[vii] S.J. Luke and D.E. Archer, "Gamma Attribute Measurements – Pu300, Pu600, Pu900," Proc. INMM 41st Annual Meeting, New Orleans, LA (2000).

[viii] T.E. Sampson, T.A. Kelley, and D.T. Vo, "Application Guide to Gamma-Ray Isotopic Analysis Using the FRAM Software," Los Alamos National Laboratory report LA-14018 (2003).