SAND2014-15306C

Prototype Hardware and Software for the Secure Branching of Facility Instrumentation

Maikael Thomas, George Baldwin, Ross Hymel, and Jay Brotz
Global Security Programs, Sandia National Laboratories
Albuquerque NM USA
mthomas@sandia.gov; gtbaldw@sandia.gov; rwhymel@sandia.gov;
jkbrotz@sandia.gov

ABSTRACT

The Enhanced Data Authentication System (EDAS) is a concept for branching measurement data from existing operator-owned instrumentation to a safeguards inspectorate. Requirements of both the facility operator and the safeguards inspector dictated the design and development of EDAS. EDAS does not modify or otherwise affect the original instrumentation signal, even with a loss of power or other failure. EDAS generates a bit-by-bit replica of the instrumentation signal that is pushed out over a separate "branch" to the safeguards inspector. The branched signal is digitally signed and encrypted using cryptographic algorithms approved by the U.S. National Institute of Standards and Technology. A branched signal originates from an EDAS junction box, 9.5 cm x 6.0 cm x 4.0 cm, inserted in the operator's instrumentation signal line, close to the sensor. So as not to interfere with the original instrumentation signal line, various means are used to ensure effective isolation of the EDAS branch. Within the EDAS box, the operator's signal line is continuous, but the device senses bidirectional digital data on the line through capacitive coupling. EDAS incorporates commercial-off-the-shelf (COTS) hardware components and open-source software; a BeagleBone Black embedded processor and the Ubuntu Linux operating system comprise the core platform driving the EDAS branch. The processor runs custom software that compiles the sensed data into packets, signs and encrypts each packet, and sends the packets using an Ethernet over USB network connection to a computer monitoring the branched data. EDAS client software, installed on the monitoring computer, receives these packets, decrypts, and authenticates the data. Flexible configuration of the branching software permits EDAS to accommodate varying data rates and burst characteristics of different sensors. Fault tolerance enables automatic recovery from system errors like loss of power or network connectivity. Prototype units have been built and software developed to operate the branching system. Initial development testing at Sandia National Laboratories is complete. We are now preparing for field testing EDAS under a joint collaboration with the European Commission, both the Directorate-General for Energy (Luxembourg) and the Joint Research Centre (Ispra). The work is supported by the NNSA International Nuclear Safeguards and Engagement Program.

Key Words: branching, measurement, operator, safeguards, authentication, encryption

1. INTRODUCTION

We seek a technical means to utilize operator instrumentation to supplement fully independent safeguards measurements, which may increase inspector confidence in being able to maintain continuity of knowledge of a facility. The Enhanced Data Authentication System (EDAS) generates a copy of an existing instrumentation signal, adds a time-stamp, cryptographically signs and encrypts the copy, and sends it over a separate branch to a passive observer (e.g., an inspector). The original conceptual basis for EDAS had been demonstrated using a PC104 computer running embedded Windows XP.[1] We have continued the development.[2] This work describes a fully revised EDAS implementation. EDAS now features a smaller form factor, uses commercial off-the-shelf components, incorporates capacitive sensing of the instrumentation signal, has been rewritten in Java for an Ubuntu Linux operating system, and has improved cryptography. EDAS is now undergoing extensive testing.

2. DESCRIPTION

2.1. System Concept

EDAS taps a copy of the digital information from existing operator instrumentation at a “branch point” close to the sensor. A passive observer, the inspector, is able to view and record the information copy sent over the “EDAS branch.” The system concept is illustrated in Figure 1.

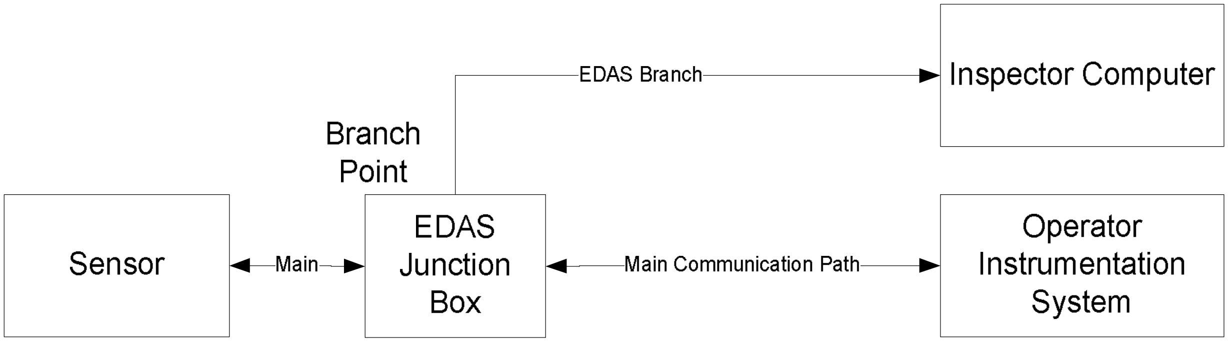

Figure 1: EDAS System Block Diagram

The main communication path is unaffected. From the perspective of the facility operator, the measurement system operates exactly the same whether or not the EDAS is present, and whether or not it is operating or powered. While operating, the EDAS branch delivers a complete copy of the information on the main communication path. It also time-stamps, digitally signs, and encrypts the information that is sent in packets to the inspector computer. EDAS assumes a bidirectional measurement system, and therefore branches both the “send” and “receive” channels in serial communication. By design, EDAS is not able to impress a signal artificially onto the operator instrumentation line.

2.2. Physical Description

At the branch point, EDAS inserts a small junction box, 9.5 cm x 6.0 cm x 4.0 cm in the existing facility instrumentation line. Presently EDAS works with an RS-232 or RS-485 signal; the junction box is equipped with two DB-9 connectors for the operator’s instrumentation line: one connecting to the sensor and the other connecting to the operator instrumentation. Each component signal line in the RS-232 is continuous within the EDAS; the galvanic path between connectors is made on a custom printed circuit board (PCB), as pictured in Figure 2. EDAS senses the digital signals on these individual signal lines capacitively, and buffers a copy of the data on the PCB to begin the EDAS “branch.”

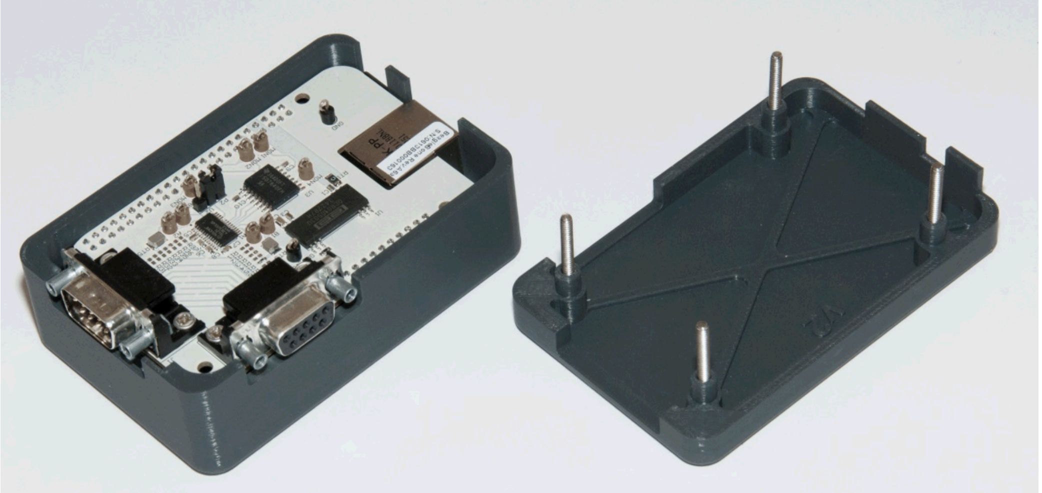

Figure 2: Opened EDAS junction box showing custom printed circuit board

The custom PC board mates to a low-cost, small form-factor, commercial off-the-shelf processor: the BeagleBone Black.[3] Note that the processor works only on the copy of the instrumentation data, not on the original signal(s) that pass between sensor and operator instrumentation. After processing, the EDAS branched information is sent to an inspector computer. The BeagleBone Black is equipped with both physical USB and RJ45 connectors; we employ Ethernet over USB to convey the branched signal.

The BeagleBone Black requires power for processing data on the EDAS branch. Note that power is not required to maintain the original operator instrumentation line, which continues to function passively as long as it remains physically connected. Power for EDAS processing is provided either over a dedicated power connection, or via USB. Since the EDAS signal branch uses the USB connection, we also use that same USB connection for power to the EDAS junction box.

Both the BeagleBone Black and the attached custom PCB (referred to as a “Cape” in BeagleBone parlance) fit inside a custom plastic enclosure. Eventually the EDAS junction box enclosure could be replaced with a tamper-indicating enclosure, planned for a subsequent development phase. Figure 3 is a picture of the EDAS junction box, identifying the external connections to the housing.

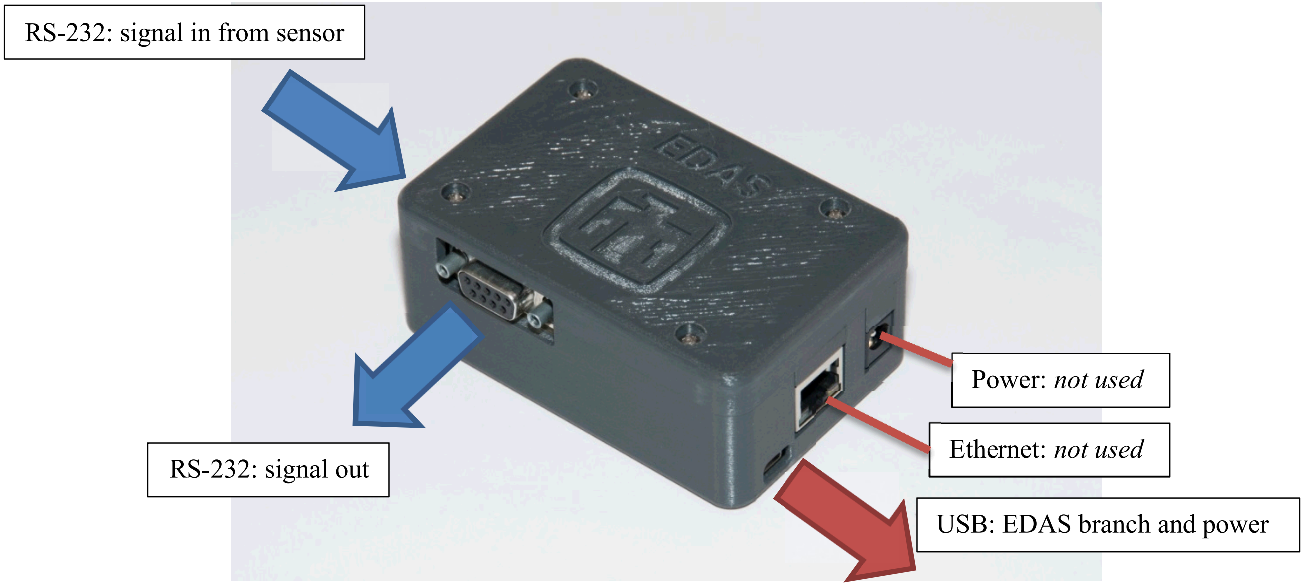

Figure 3: EDAS Junction Box

2.3. Software Description

Custom software running on the BeagleBone Black implements the EDAS branching functionality, described in the next section. EDAS software is written in Java, to encourage its use on multiple hardware platforms and operating systems. The EDAS junction box runs an open source version of the Ubuntu Linux operating system. The operating system boots automatically when power is available over the USB connection; the operating system then starts the EDAS software automatically. The software can immediately and asynchronously branch input signals.

An inspector computer, attached to the EDAS branch from the junction box, has custom software to receive the secure branched data from the EDAS junction box. The software is currently compatible with a Windows 7 operating system. The inspector computer software is capable of beginning the data acquisition at any time. Both the operating system and EDAS software can be scripted to start automatically when the computer has power.

Cryptographic modules from the open source BouncyCastle cryptographic library[4] digitally sign and encrypt data on the EDAS branch. The software on the inspector computer uses routines from BouncyCastle to decrypt and authenticate the branched data.

2.4. Functional Description

Software in the EDAS Junction Box collects and buffers all data transferred over the main communication path, compiles the data into discrete packets, adds a time stamp to each from an on-board clock, digitally and encrypts the packets, and then pushes them over a TCP connection to the inspector computer. The software in the EDAS Junction Box also periodically creates and sends heartbeat messages to confirm that EDAS is operating normally.

EDAS has no a priori understanding of (or expectation for) the data it branches. It needs the flexibility to branch data streams that could appear in relatively short bursts (the triggering of a bar code reader, for example), or that run continuously (e.g., the output of a laser scanning rangefinder). In the first case, the ideal situation would be to keep all of the functionally-related data together in the same packet. In the second case, the stream is split into contiguous blocks (individually signed and encrypted) that must be reassembled later. To accommodate both extremes of data situations, EDAS uses configurable size and time "limits" in constructing data packets. The EDAS packet builder logic works with *both* size and time limits; the logic is illustrated in Figure 4.

The size limit can be set as either "hard" or "soft." A hard size limit defines a *maximum* data packet size, in number of bytes. A soft limit defines a *minimum* data packet size, in number of bytes. Once enough data bytes have accumulated to reach the hard or soft limit, EDAS will create the packet and send it over the network to the inspector computer. In either case--hard or soft size limit--the data packet is also subject to a time limit. The time limit is how long (in milliseconds) EDAS will accumulate data for a single packet. If the time limit is exceeded before the size limit is reached (hard or soft), EDAS will create and send the packet to the inspector computer.

EDAS digitally signs all data (including heartbeat messages) transferred over the EDAS branch via TCP connection. It generates a private/public key pair using a random source generated by noise from operating system input/output ports. The firmware stores the private key locally on the EDAS Junction Box, and sends the public key to the client software on the Inspector Computer upon establishing a connection. Only the server will be able to sign packets using the private key; the client uses the public key to verify that the messages are authentic. The EDAS software signs all messages using k-283 Elliptical Curve Digital Signature Algorithm (ECDSA)[5], which uses a Koblitz curve over a binary field with 283-bit parameters.

EDAS also encrypts all data (including heartbeat messages) transferred over the EDAS branch via TCP connection. EDAS uses the Advanced Encryption Standard (AES), a symmetric key algorithm, for both encryption and decryption. The key length is set to 128 bits. The EDAS junction box and Inspector Computer use a Diffie-Hellman key exchange to establish a shared AES key securely.

3. DISCUSSION

3.1. Isolation of EDAS from operator line

The EDAS junction box has been designed so as not to interfere with the operator signal line in any way. Minimal electrical circuitry is attached to the operator signal lines. A tap-off runs through a current-limiting resistor to a transceiver, which converts the RS-232 (or RS-485) signals to levels understandable by EDAS and passes the result through an isolation barrier. The barrier is a digital isolator, Texas Instruments part ISO7640FM, [6] which ensures that the operator-side instrumentation and the EDAS-side electronics are separated galvanically. Testing has assured us that if the transceiver shorts, either to ground or to the supply voltage (3.3V), the RS-232 signal would still be recognized on the operator instrumentation line. Also, all power provided to the EDAS operator-side electronics is isolated with a magnetic power supply, Texas Instruments part DCR010503.

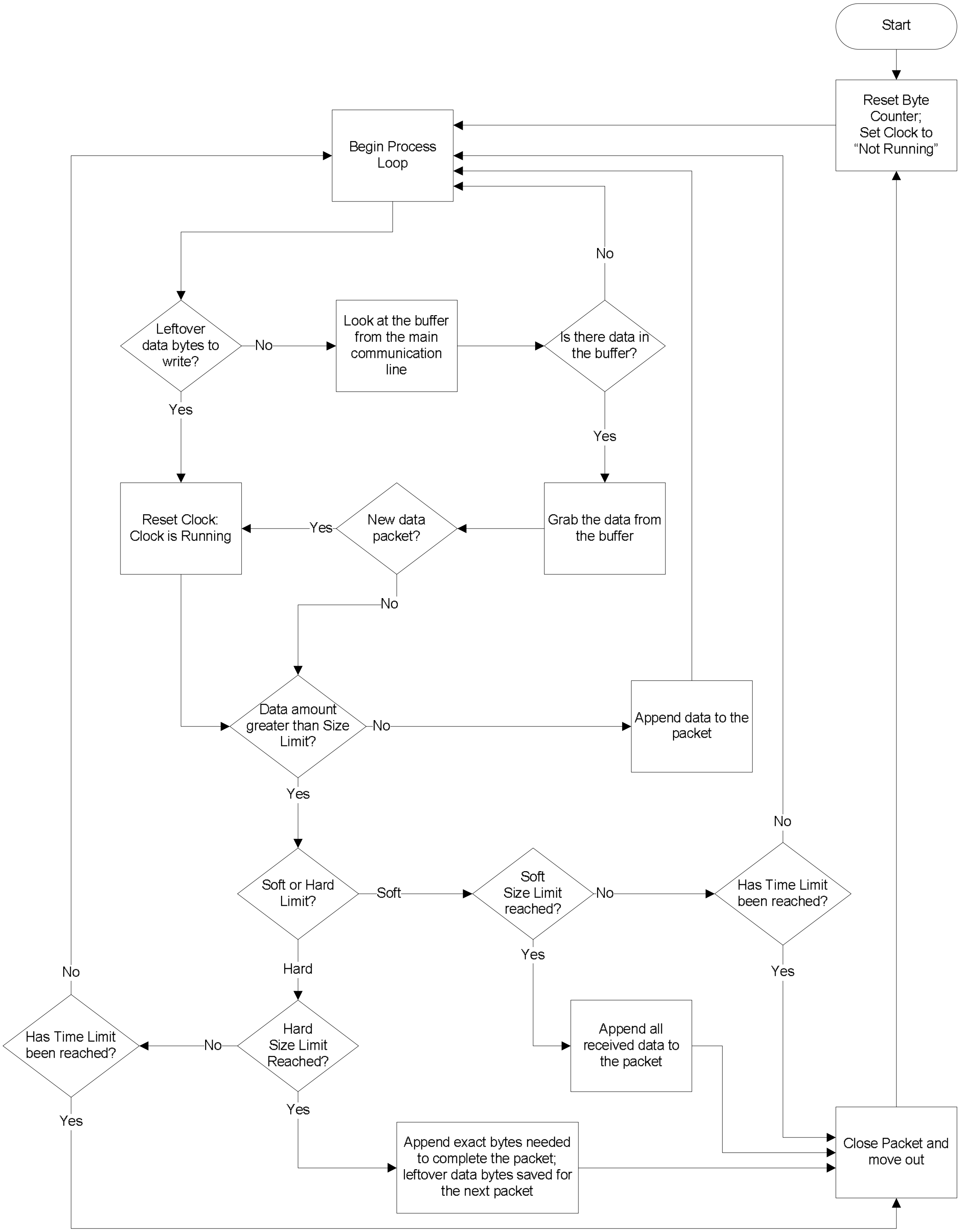
Start
Reset Byte Counter; Set Clock to “Not Running”
Begin Process Loop
Leftover data bytes to write?
No
Look at the buffer from the main communication line
Is there data in the buffer?
No
Yes
Yes
Grab the data from the buffer
New data packet?
Yes
Reset Clock: Clock is Running
No
Data amount greater than Size Limit?
No
Append data to the packet
Yes
Soft or Hard Limit?
Soft
Soft Size Limit reached?
No
Has Time Limit been reached?
No
Yes
Yes
Append all received data to the packet
Hard
Hard Size Limit Reached?
No
Has Time Limit been reached?
No
Yes
Yes
Append exact bytes needed to complete the packet; leftover data bytes saved for the next packet
Close Packet and move out

Figure 4: EDAS software logic used in forming data packets

This built-in isolation protects the operator signal line from unexpected errors or failures that may arise from the EDAS electronics. Just as important, this isolation ensures that no signal could be impressed onto the operator signal lines from the EDAS branch. Specifically, each data isolation channel is connected to the isolation barrier via a logic input and output buffer. These buffers enforce directionality, preventing noise currents and other sources of interference from entering the operator signal lines. Likewise, the isolated power supply provides the same protection to the local ground of the operator instrumentation. Non-interference is intended to be an inherent feature of EDAS, assuring an operator that EDAS presents a low risk of affecting facility operations adversely.

3.2. Fail-safe operation

In consideration of operator requirements, operator signal connections do not depend on the state of the EDAS junction box. Whether the junction box is powered off, powered on, or in the process of turning on or off should make no difference to the state of the operator lines. Even an unlikely but catastrophic failure on the EDAS side, such as a high voltage spike, would be blocked from entering the operator signal path.

Should a facility operator have any concerns whatsoever, the connections have been designed such that the operator line can be physically unplugged from the EDAS Junction Box, and the two ends of the operator signal line connected directly to one another, bypassing EDAS altogether. Other than not being able to sense any operator data at all, EDAS is not currently designed to detect whether an operator cable has been disconnected.

3.3. Accurate, complete, and meaningful branched data

In consideration of inspector requirements, the EDAS junction box captures a bit-by-bit and complete replica of the bi-directional data on the operator instrumentation signal line. As discussed above, EDAS has no understanding of the data that it is branching, which greatly simplifies its design. Packet formation blindly follows a prescriptive logic but with configurable time and size limits; if suitably chosen, those limits can result in packets that replicate the native packets of the instrumentation. But even if these parameters are not chosen optimally, EDAS packets will still be faithfully reassembled by the inspector computer to recover the original data stream. A post-processing step, apart from EDAS, is necessary to interpret meaningful information from the branched data.

3.4. Data confidentiality and authentication

Cryptographic authentication is important to identify the source and to protect the integrity of data transmitted between the EDAS Junction Box and the Inspector Computer.

Encryption is important to keep the data transferred on the TCP network connection confidential. “Safeguards confidential” respects the privacy of the shared information, and prevents an eavesdropper from obtaining the information that the operator has shared with the inspector in confidence. (Encryption does not prevent an eavesdropper from discerning whether or not EDAS is functioning, however.) The Inspector Computer would need to have additional security to preserve the confidentiality of the branched data after decryption.

3.5. Next Steps

The EDAS prototypes are currently undergoing functional, system and integration testing. Although Sandia, as the developer, has tested EDAS operation extensively, we depend also on outside testing and evaluation for an independent confirmation that EDAS meets both the operator and inspector requirements. Both the European Commission Directorate General for Energy (Luxembourg) and Joint Research Centre (Ispra) are currently testing EDAS hardware and software to ensure readiness for an eventual field trial. Results from the testing and field trial will be reported in an upcoming paper.

4. CONCLUSION

EDAS ensures that the branched information is a secure, true, and complete replica of the data streams with the primary instrumentation. At the same time, EDAS is designed to minimize any additional risk to the measurement system from branching. EDAS could enable an international nuclear safeguards inspectorate to view process data from instrumentation belonging to a nuclear facility operator. Such data would complement the information obtained from safeguards measurements that are fully independent from those of the operator. EDAS is currently undergoing testing and integration with inspector safeguards systems, in anticipation of a future field trial.

ACKNOWLEDGMENT

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

SAND2014-xxxx x

REFERENCES

1 João G.M. Gonçalves, et al; *Enhanced Data Authentication System (EDAS): Concept, Demonstration and Applications*; proceedings of the Institute of Nuclear Materials Management 52nd Annual Meeting, Palm Desert, California, July 2011.

2 Maikael Thomas, et al; *Enhanced Data Authentication System: Converting Requirements to a Functional Prototype*; proceedings of the European Safeguards Research & Development Association 35th Annual Meeting, Bruges, Belgium, May 2013.

3 http://beagleboard.org/products/beaglebone%20black

4 http://www.bouncycastle.org/java.html

5 http://csrc.nist.gov/groups/ST/toolkit/digital_signatures.html

6 http://www.ti.com/product/iso7640fm