10.2478/v10048-010-0012-8
MEASUREMENT SCIENCE REVIEW, Volume 10, No. 2, 2010

Generic System for Remote Testing and Calibration of Measuring Instruments: Security Architecture

M. Jurčević, H. Hegeduš, M. Golub

Faculty of Electrical Engineering and Computing, Department of Electrical Engineering Fundamentals and Measurements, University of Zagreb, Unska 3, 10000 - Zagreb, Croatia, {marko.jurcevic;hrvoje.hegedus;marin.golub}@fer.hr

Testing and calibration of laboratory instruments and reference standards is a routine activity and is a resource and time consuming process. Since many of the modern instruments include some communication interfaces, it is possible to create a remote calibration system. This approach addresses a wide range of possible applications and permits to drive a number of different devices. On the other hand, remote calibration process involves a number of security issues due to recommendations specified in standard ISO/IEC 17025, since it is not under total control of the calibration laboratory personnel who will sign the calibration certificate. This approach implies that the traceability and integrity of the calibration process directly depends on the collected measurement data. The reliable and secure remote control and monitoring of instruments is a crucial aspect of internet-enabled calibration procedure.

Keywords: internet-enabled testing and calibration, travelling standard, hardware security module (HSM), ISO/IEC 17025

1. INTRODUCTION

CALIBRATION and testing of measuring instruments is an important part of the metrology activities. Also, it is a routine operation. All laboratory instruments require recalibration in regular time intervals. Traditionally, instruments are then transported to the calibration laboratory which implies a downtime for every particular instrument under calibration.

The development of automated measurement instruments, evolution of the computer networks and growing number of quality standards in the industrial and research fields in recent years has led to the realization of internet-enabled calibration systems, specifically for industrial applications [1] [2].

Internet-enabled metrology as a wider concept is a term that covers the use of telecommunication systems to provide convenient access to a range of measurement and calibration services [1].

These services usually include some of the following:

- remote control and monitoring of measuring equipment (instruments and standards),
- traceable measurements that are performed at a customer location but controlled remotely by the calibration facility (this covers the term internet-enabled calibration and/or testing),
- access to measurement and calibration history and other related data of each instrument and standard,
- access to libraries of testing and metrology software or algorithms [4] [5].

Remote testing and calibration systems also provide new possibilities for the National Measurement Institutes (NMIs), because measurement procedures can be done more rapidly and more securely for the instrument or standard that needs to be calibrated, even resulting in increased accuracy of the calibration.

Remote calibration approach is especially efficient whenever the calibration of low-cost instruments is involved. The calibration of these instruments is generally required by a number of quality standards, and is quite expensive when compared with the instrument cost. From the other point of view, since this kind of equipment does not require top-class calibration devices, neither demands ambient conditions, it can be performed by using travelling standards.

The remote testing and calibration of measuring instruments is an interesting application that is still not fully exploited, mainly due to the legal issues because of the possible lack of security, which is associated with the operations outside of a calibration laboratory.

A number of security issues are involved, since the installation should be done by the customer staff and under some circumstances may not be under total control of the calibration laboratory personnel who will sign the calibration certificate.

This approach implies that the traceability and integrity of the calibration process (that is in this case performed over some communication network) directly depends on the collected measurement data. The reliable remote control and monitoring of instruments is a crucial aspect of the internet-enabled calibration procedure.

This paper proposes on-site calibration service, which is based on a travelling standard and a multi-layer network application. This service allows remote execution of the calibration procedure, automatic acquisition and fast processing of calibration results.

2. GENERAL DESCRIPTION AND PROPOSED APPROACH

Many of the commercially available software tools devoted to the implementation of Virtual Instruments (VIs) (e.g. LabView LabWindows/CVI™ and HP VEE™) offer features for interactive control of instrumentation via the Internet and in that way testing and calibrating some devices remotely. These programs are based on the idea of VIs as an abstraction of hardware and software functions of every represented instrument. These proprietary solutions usually require special run-time components to be installed on client computers and are usually not freely distributed.

The main goal of the proposed internet-enabled calibration system [2] called *iCal* is to enable the control and supervision

of the remote standard(s) and instrument(s) that are used in a calibration process under the conditions that are defined accordingly to the international standard ISO/IEC 17025:2005 [3].

This system consists of a PC-manageable travelling calibration device or artifact standard (travelling unit, TU) that is sent to the client laboratory and a CSP (Calibration Service Provider) server-side application system that controls and monitors the whole process of calibration. It can be any type of calibration device, for example digital multimeter or signal generator.

Travelling standard and device/unit that are under test and/or calibration (UUT) must have a communication interface (e.g. USB, LAN or GPIB) in order to connect them to the client PC that controls the calibration event.

Application on the CSP side performs the calibration procedure without any human assistance on the client side. Also, client-side equipment (CSE) has to self-recognize and make automatic configuration of available interfaces, connected instruments and standards.

After establishing a connection between server and client using ordinary web browser applications, successful authorization using a previously issued personal smart-card, Java applet will open the main form and the remote user is presented with a user interface showing a list of available interface cards and instruments connected to them. According to user's rights in the calibration software, the user can select a specific calibration procedure from a list of available ones, clicking on its name. Before the calibration starts, the software starts a sequence of tests on IEEE 488 interface and instruments physically connected to it. Simultaneously, the client part of the application returns information about hardware status to the server.

Server returns a set of instrument-specific methods needed to drive all the hardware for the specific calibration event to the client side and initiates calibration sequence by sending commands to the client part of the application. The client-side calibration system (which now consists of both CSE and UUT) receives all the commands and instructions from the CSP server. It makes continuous scanning of the instrument readings and sends those results to the server on the CSP side.

All the relevant operations required to create the calibration certificate (data storage, processing and calculation) are executed on the CSP side. On the completion of the calibration procedure, the results are processed, stored and also returned to the client and displayed.

After the calibration procedure is completed, all the client-side calibration system components (CSE) are returned to the CSP office for a test and verification. If the equipment passes this test, a calibration report is sent to the client.

This document should contain the date and time of calibration event, some information about calibrator output data, the minimum and maximum measurement uncertainty, the measurement results, the differences between measured data and calibrator outputs, and the calibration timing.

The use of a client-server application is mandatory in order to control the calibration procedure and to perform a real-time check of the calibration results. This makes the operator on the CSP side able to address error events that could happen while the procedure is in progress. On the other hand, such a choice requires protection of sensitive information that is sent over the Internet. Also, this operation is not in total control of the CSP staff, so the client-side PC and calibration management software can be illegally modified before the calibration procedure starts.

3. SYSTEM COMPONENTS DESCRIPTION

A framework for a generic PC-based calibration system that is able to adapt to different types of calibrations is proposed. The solution consists of a hardware (calibration instruments and additional hardware equipment on the client side) and a software part (server and client modules) [2].

Model of the client-side equipment – the travelling unit (TU) is made up of one or more reference standard(s) or device(s) equipped with a communications module.

The communications module is a common name for a component that communicates with the client-side control PC. For this purpose an IEEE 488 internal or external (USB, Ethernet, RS-232 or other type) interface card is usually used. The communications module sets all the parameters for reference standard and collects the measurement data during the verification and calibration process. A calibrated environmental probe can be used if necessary to acquire the client-side temperature, pressure and humidity values.

The calibration procedure consists of:

- identification and configuration of remote side equipment,
- identification and verification of UUT (using the IEEE 488 communications and visual control if necessary),
- testing the functionality of the remote calibration system,
- acquisition of the measurements carried by reference standard (or device) and UUT.

Regarding the software part, the calibration system architecture consists of server and client unit. The implementation code is divided into several main parts (as shown in Fig.1 and Fig.2):

- a set of instrument-specific calibration scripts that allow to update or add new programs for measurement procedures,
- a device-independent control and monitoring layer,
- network communications control layer,
- database storage management system layer.

Architecture mentioned above assures several advantages: simplified management and control of the hardware resources, reliable and secure communication between client and CSP side and foolproof software upgrades and modifications.

Most of the system operations that are carried out within a test and calibration procedure are based on the abstract functional layer. This approach helps to simplify software control and monitoring of instruments, because it is independent of the instruments connected to the client-side control computer.

The software that runs CSP and client-side calibration and certification procedures is under constant development. The core software components that run on the client side depend on Java technology. Java ensures portability and permits usage of Java Native Interface (JNI) providing direct interaction to

the native instrument drivers usually distributed from their manufacturers as C++ dynamic link libraries (C++ DLLs). This way it is convenient to drive hardware peripherals and instruments.

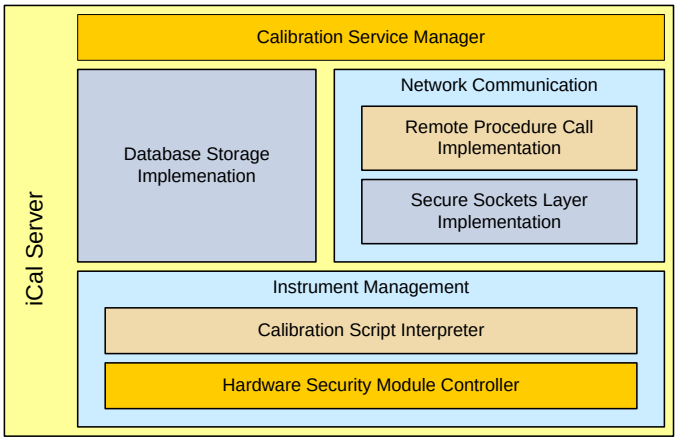

Fig.1 Architecture of *iCal* server

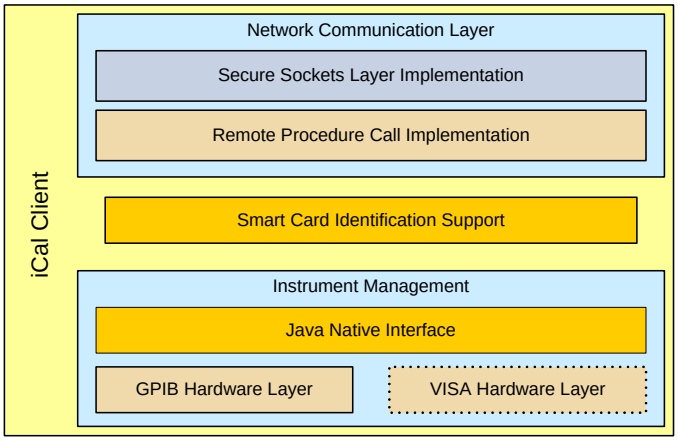

Fig.2 Architecture of *iCal* client

Client-side software components are divided into several main parts (as shown in Fig.2):

- platform-dependent driver interface to the communications controller (e.g. GPIB interface card) manufacturers drivers (one DLL library under Windows or SO library under Linux),
- platform-independent Java interface to the platform-dependent driver interface (in a form of a JAR file),
- main application that provides GUI to the client personnel and operates the calibration procedure in a form of signed and authorized JAR (Java ARchive) applet or JNPL (Java Web Start) application.

Also, client-side software needs GPIB controller (or other type of interface controller) drivers to be installed on the client-side PC.

Java applet [2] that runs on the client side is relatively small in size (approx 50 kB in size) and has no dependency on a specific calibration procedure at all.

It is actually a small graphical user and hardware interface that is controlled from a server side over Java Remote Procedure Call (Java RPC). It receives all the commands and sends measurement results over custom-designed RPC protocol (that is transported over network using secure connection).

Custom RPC message and data structure as a DTD (*Document Type Definition*) is as follows:

```
<!DOCTYPE rpc [
<!ELEMENT rpc (function+)>
<!ELEMENT function (item*)>
<!ELEMENT item (#PCDATA|item)*>
<!ATTLIST rpc version CDATA #REQUIRED>
<!ATTLIST function name ID #REQUIRED>
<!ATTLIST item name CDATA #REQUIRED>
<!ATTLIST item type (string|integer|float|struct) #REQUIRED>]>
```

This is generic RPC message interchange definition that is used for sending function requests and data transfer between server and a client. It provides possibility for extensions and upgrades of functions and new control commands in future without the need for client code modifications.

Fig.3 shows the current version of the authorization protocol and data transfer between server and client part.

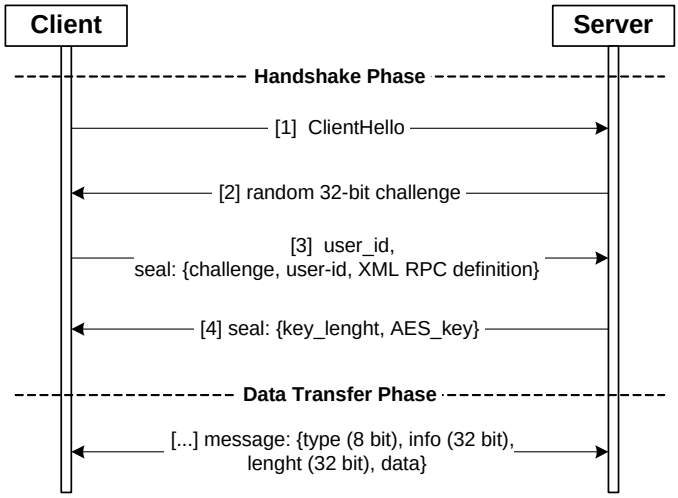

Fig.3 Authorization protocol between client and server

Authentication and secure network connection relies on a pair of public and private keys (public-key infrastructure, PKI) assigned to calibration system server and all the users and instruments involved in testing and calibration procedures [6] [8]. Private keys are placed on smart-cards and are distributed to personnel that operate the calibration system (and calibration equipment). Public keys belonging to them are kept in the directory services server database.

There are several steps involving secure network connection:

1. client connects to a server over some kind of telecommunication network,
2. server generates challenge data (32-bit) and sends it to a client,
3. client returns digital seal using RSA algorithm based on challenge code received from server, user identification (32-bit) and definition of supported RPC protocol in a XML form (server's public key is already stored on a user smart-card),

4. server checks received information – if a user is successfully authenticated, another seal is generated with AES (*Advanced Encryption Standard*) key information. AES encryption protocol is used for symmetrical encryption of all the transferred data in a calibration session.

Instrument and calibration management programs are created based on manufacturers programming manuals of each instrument and are dynamically loaded on the server side according to the connected equipment and instruments on the client side.

These programs are used to generate client-side instrument commands and interpret returned data. Instrument management layer assures transport of commands, control messages and data to the connected measurement equipment. The whole core of the software application (client-side program component and instrument drivers) resides permanently on the server. This way, there is no need to install any special software tool on a client PC, except for a Java Virtual Machine runtime (JVM) and placing two files mentioned above into operating system directories (which is done automatically during the start-up of a client-side application), providing thin-client application architecture safety for the whole calibration system.

All the control information, calibration results and raw measurement data between server and client side is sent using TCP/IP protocols (usually Internet or some type of private network) to the Calibration Service Provider (CSP) laboratory, where they are stored in a database. These results are also stored on a client-side control PC or, if possible, in a TU memory.

As soon as the TU and control PC return to the CSP, the data received through the Internet is compared with the one stored on the control PC and/or TU. If both sources of calibration data are identical and the TU is within its specifications, the laboratory issues the calibration certificate of the calibrated device that can be then delivered to the client.

3.1. Hardware security issues of remote calibration

There is a potential problem, usually neglected or ignored because of insufficient understanding, but requiring much consideration. It is the communication between client-side PC and other instruments, for example over the often used IEEE 488 – GPIB interface. Since this protocol has no built-in security elements and was not designed to support integrity, encryption and confident transmission, communication over this media is endangered.

The main risk points of unauthorized access and alteration of the calibration system are the data (e.g. GPIB) connections between the client-side PC and CSE and UUT.

For example, someone can use specially designed software components to listen to and intercept GPIB commands and data transmissions, altering them, everything leading to forged calibration data.

The standard IEC 60488-2 [7] defines a set of commands that every measurement instrument should implement. One of the mandatory commands is "*IDN?" that implements identification query of every device over the GPIB system interface. Every instrument should respond to this command with the following information:

- full manufacturer name,
- instrument model,
- serial number,
- firmware version or equivalent data.

With this information calibration software could get enough information to uniquely identify, to be able to self adopt and distinguish instruments and other connected equipment to the GPIB bus.

Unfortunately, there are a lot of digital instruments in use that have the GPIB interface installed, but with very limited firmware implementation of "*IDN?" command. For example, a lot of digital multimeters on the market respond to this command with no identification data at all, but with current measured data values.

Another serious problem occurs in a situation when a large number of instruments are remotely calibrated. One could, by chance or intentionally, mix up instruments or place the same instrument in the calibration process several times, leading to an invalid calibration certificate.

In these cases, from the obvious reasons, it is not possible to leave preparations for remote on-site calibration to the client laboratory staff, thus limiting remote-enabled calibration process advantages.

To minimize such possibilities, at least the information between CSE and client-side PC should be encrypted. The decryption process would then be performed within the CSE.

Also, the measurement connection between UUT and CSE can be compromised. As an example, there is no way how to prevent possible connection of the multimeter inputs to some other voltage source able to generate voltage levels that could potentially lead to false positive testing results.

These risks may be lowered if the surveillance camera is used (when appropriate). It could provide some kind of visual control for the equipment involved in the calibration process.

3.2. Model of hardware security module

As shown above, currently the GPIB, as a frequently used protocol for interconnection and control of measuring instruments, but also other communication interfaces and protocols does not support basic security techniques such as:

- authentication,
- integrity,
- confidentiality,
- nonrepudiation and
- access control

of all the data and instruments involved in a measurement process.

This leads to a number of problems regarding remote testing and calibration of instruments.

The GPIB is basically a simple plain-text message protocol with no security services implemented. It allows any form of passive and active eavesdropping where it is possible to track or even intercept and change existing or inject new messages transferred between measuring instrument and a PC that controls the calibration process. It is also possible to impersonate another or existing measuring instrument and return invalid or forged measuring results.

Of course, this is not the case in a regular calibration

laboratory where all the measurement procedures are done under total control of the laboratory staff. But, in a case of remote testing and calibration, someone with (financial?) motivation, enough experience and appropriate hardware could make the above-mentioned attack leading to an invalid calibration certificate.

The main aim for the proposed generic system for remote testing and calibration is to solve these open questions.

Beside the client and server software components that include the abovementioned common security mechanisms, a special hardware security module (HSM) was designed as a prototype model for possible future GPIB protocol updates.

The block diagram of the HSM module is shown in Fig.4. The basic idea was to upgrade the current version of GPIB protocol used for needed security services. As it was not possible to modify firmware of existing measurement instruments, an additional device was developed. It serves as an interface between an existing GPIB interface integrated in a measuring instrument and a GPIB bus used for connection to the calibration controlling PC. It is uses the described HSM paired with client software running on the PC that is able to transfer encrypted GPIB data from server side.

To be able to fulfill its purpose, the HSM is supposed to be connected and sealed to the GPIB interface of a measuring instrument. In this case, it is possible to regard HSM as a part of an instrument. Second GPIB port is connected to the GPIB bus interfacing other instruments and PC that controls the calibration process. HSM is able to pass-through regular GPIB traffic, but it also implements an expanded set of new GPIB commands providing encryption of traffic between instrument and PC. These commands are used during the test or calibration process using *iCal* system.

The HSM is based on an ATMEL ATMega128 8-bit microcontroller (MCU) with additional components linked to the GPIB interface.

Implemented algorithms for symmetrical and asymmetrical encryption/decryption are based on Atmel AVR *AVRCrypto* library available from [9].

The current prototype of an HSM described here has a pair of public-private keys stored in its internal program memory.

This is a potential security risk as it is possible to read the microcontroller's memory contents, thus the private key content (but the private key must remain secret, not known to anyone). This should be solved in future revisions of the HSM, placing key pairs on a smart-card or similar highly-secured environment.

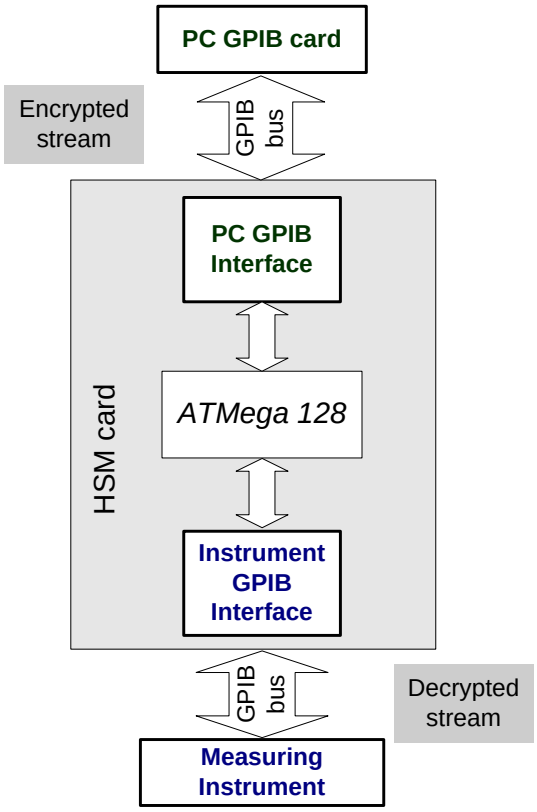

Fig.4 Block diagram of the HSM module

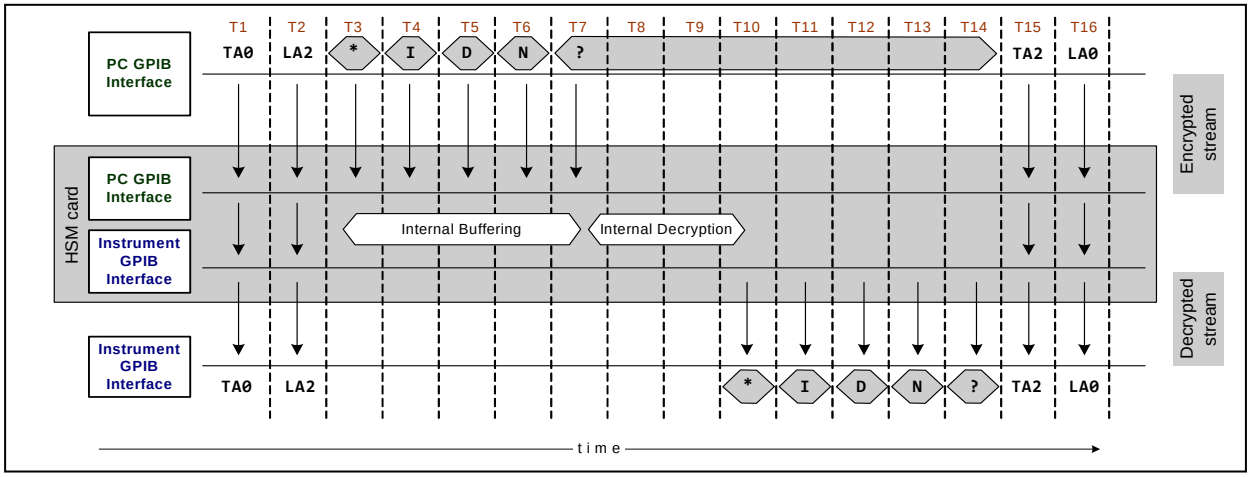

Fig.5 Data transfer between PC and measuring instrument equipped with HSM card

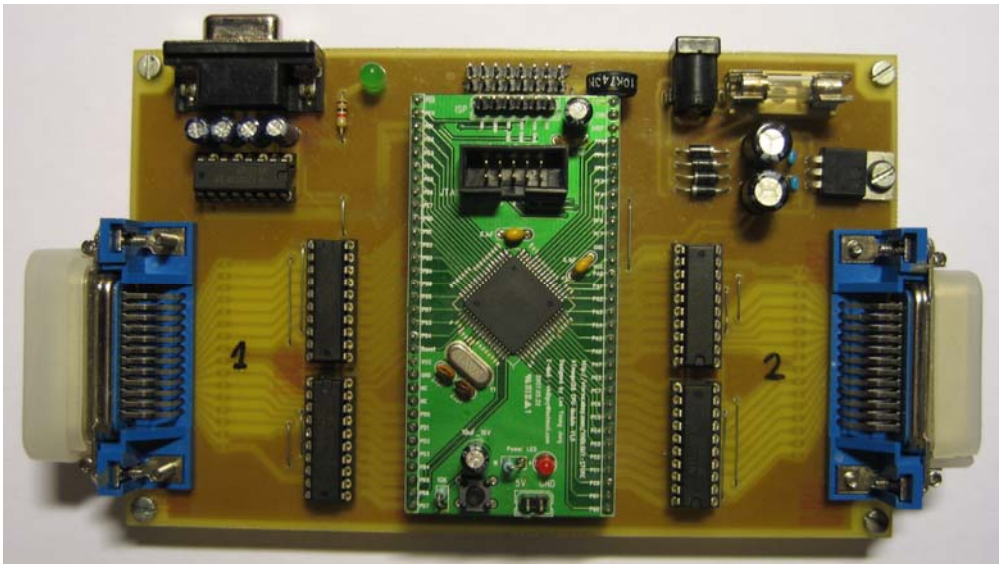

Fig.6 Working prototype of HSM card

Fig.5 shows an example of modified timings on the GPIB bus regarding synchronization of control and data lines during transmission of encrypted command for device identification (GPIB command *IDN?) between PC and instrument. Similarly, the response from the instrument would flow in the opposite direction. Only the addressing and data information is shown on the diagram, other control commands were omitted.

The timings on the GPIB bus are slightly changed because of buffering (and possible encryption/decryption process) in the data interchange process between instrument, HSM and a PC. This leads to speed degradation of the GPIB bus in secured data transfer mode to the average of 10 kBps. Use of a faster MCU would speed up the rate of data transfer.

Fig.6 shows a working prototype of the HSM card. These units were used only for research purposes. Of course, for daily test and calibration usage in a real laboratory environment, it is necessary to develop a much smaller unit.

In the future, the IEEE 488 and similar, future coming instrumentation interfaces and technologies should be upgraded for security services, providing the same protection level as used in computer network surroundings.

4. CONCLUSIONS

It is possible to imagine many different remotely enabled calibration procedure implementations, but not every calibration is equally suited for remote operations. Some examples of a successful application of remote calibration procedures can be found in [2] and [10].

One of the leading prerequisites is that a calibration procedure is highly automated and that only a minimum of human interaction is required. Usually it is needed to have, for example, a multifunctional calibrator for a calibration of electrical quantities, or extra hardware that will automate and facilitate the measurement process, keeping human interaction at minimum level.

Complex numerical analysis of measurement results and knowledge that stands behind are important parts of a calibration process. In case of a remote calibration process, all the expertise and knowledge can be delivered to the client more easily compared to a classical calibration process.

REFERENCES

[1] R. A. Dudley, N. M. Ridler (2003 February). Traceability via the Internet for Microwave Measurements Using Vector Network Analyzers, *IEEE Transactions on Instrumentation and Measurement*, vol. 52, no. 1, pp. 130-134.

[2] Jurcevic, M.; Borsic, M.; Malaric, R.; Hegedus, H., (2008 September). Internet-Enabled Calibration Services: Design of a Secure Calibration System, *IEEE Transactions on Instrumentation and Measurement*, vol. 57, Issue 9, p.p:2012 – 2018.

[3] ISO (2008). *ISO/IEC 17025 General requirements for the competence of testing and calibration laboratories*, http://www.iso.org.

[4] *Advanced Mathematical and Computational Tools in Metrology (AMCTM)*, http://www.amctm.org.

[5] IMEKO Technical Committee TC21: *Mathematical Tools for Measurements*, http://www.imeko-tc21.org.

[6] Stallings, W. (2002). *Cryptography and Network Security*, Prentice Hall.

[7] IEC (2004). *International standard IEC 60488-2 – IEE 488 - Standard digital interface for programmable instrumentation – Part 2: codes, formats, protocols and common commands*, http://www.iec.ch.

[8] *Understanding PKI: Concepts, Standards, and Deployment Considerations*, 2nd edition, Addison Wesley, 2002.

[9] Emile van der Laan (2008). *AVRCryptoLib v. 0.51*, http://www.emsign.nl.

[10] Malarić, R.; Hegeduš, H.; Mostarac, P. (2009). Use of Triaxial Accelerometers for Posture and Movement Analysis of Patients, *Advances in Biomedical Sensing, Measurements, Instrumentation and Systems*, (pp. 127-143). Berlin, Springer Verlag.

Received October 5, 2009.
Accepted April 6, 2010.