USB Keyboard Security Unit

Wojciech Wodo, Lucjan Hanzlik, Konrad Zawada

Abstract*— Every user has its specific rhythm of typing which could be used as a biometrics in order to build some kind of "footprint" – unique profile. If somebody gets to know this profile, legitimate user is endangered by tracking and being impersonated. That is the way typing rhythm must be protected. We designed a hardware-based device in order to protect the identity of the individual during usage of keyboard (typing). The unit is plugged between the keyboard and the personal computer and works as an interface modifying data on the fly in the model "man in the middle". Thanks to these modifications, an adversary who eavesdrops communication between a legitimate user and workstation gets practically no information about the "keystroking identity" of user. The security unit is based on two microprocessors: AVR AT90USB1287 working as USB Host - simulating workstation and AVR Atmega88 working as USB Device - simulating virtual keyboard. In the paper we present technical details of the security unit including electronic schemes and PCB referring to previously designed protection algorithms and results of performed efficiency tests as well.

Index Terms*— biometrics, security and privacy protection, microprocessors and microcomputers, user interfaces, human factors in software design.

I. INTRODUCTION

Biometrics is often used in computer systems as a mean to identify and authorize specific users. The security of such systems relies on the assumption that one can, with high probability, distinguish the biometric features of two different users. There are many types of biometric features with different usability. In this paper we focus on the typing rhythm of users. Numerous researchers have shown that the typing rhythm can be used to identify users [1]–[3]. The authors of those paper show that it is possible to create user models using the captured, from the keyboard, data and compare those models. However, there is a lack of articles concerning the privacy issues. Most of the published ones only consider the use of this biometric feature for honest identification i.e. access to some resource. However, there are papers e.g. [4] that consider the case of malicious identification. This is important since we live in times of global surveillance and there is a demand for anonymity. Users use specials tools (e.g. *The Onion Router (TOR)* network, proxy) to maintain anonymous in the Internet. However, note that the typing rhythm can easily be used as a side channel attack on those tools, no matter how good they are. In [5] we have shown algorithms that can be used to protect the keystroke identity of users. Those algorithms can easily be implemented since they only operate on raw data coming from the keyboard.

However, a software implementation comes always with a risk of malfunction (program does not start or stops working without a info to the user) or malicious modification (a virus or Trojan horse turns the protection off). Note that a hardware implementation cannot be modified remotely and in case of malfunction the keyboard stops working on its own. That is why, in this paper we will show a state of the art hardware implementation of the algorithms from [5] in form of a USB Security Unit. We will not rewrite the algorithms, however we will refer to them. The paper is organized as follows: Section 2 divided in two main subsections describing works on prototype of hardware-based unit and minimizing the device, Section 3 consisting of practical efficiency tests and its results and Section 4 in which we conclude our research and set new challenges for development.

II. HARDWARE SECURITY UNIT

In this section we present information about the device prototype, including details of implementation of the protection algorithms mentioned in the introduction. In the end of this part of the paper we show our efforts to minimize the security unit. We attach electronic and *Printed Circuit Board (PCB)* schemes.

A. Prototype of the unit

The idea of the device implementing protection algorithms is quite simple. Such a unit should intercept the signal coming out from the keyboard, modify it (i.e. delay or add to queue) and next send it to the workstation.

This solution works well for keyboards of old PS/2 connector type. In the case of new USB keyboards, communication between workstation and keyboard is much more complex (it is based on *Device Class Definition for Human Interface Devices protocol – HID* [6]). Thanks to using HID protocol we can obtain one universal keyboard driver for workstation, which is able to work with all of the keyboards implementing HID (in basic mode). All of available on the market USB keyboards use this protocol and our device has to implement it as well (in order to support keyboard just like the workstation). On the other hand, the unit has to emulate a keyboard to the workstation side simultaneously. To sum up, the device is plugged between a real USB keyboard and a workstation and works in the *Man in the Middle (MITM)* manner.

While designing we took into account two aspects: total price of the unit and ease of modification.

Manuscript received on May, 2014.

Wojciech Wodo MSc, Wroclaw University of Technology, Faculty of Fundamental Problems of Technology, Wroclaw, Poland.

Lucjan Hanzlik MSc, Wroclaw University of Technology, Faculty of Fundamental Problems of Technology, Wroclaw, Poland.

Konrad Zawada BSc, Wroclaw University of Technology, Faculty of Electronics, Wroclaw, Poland.

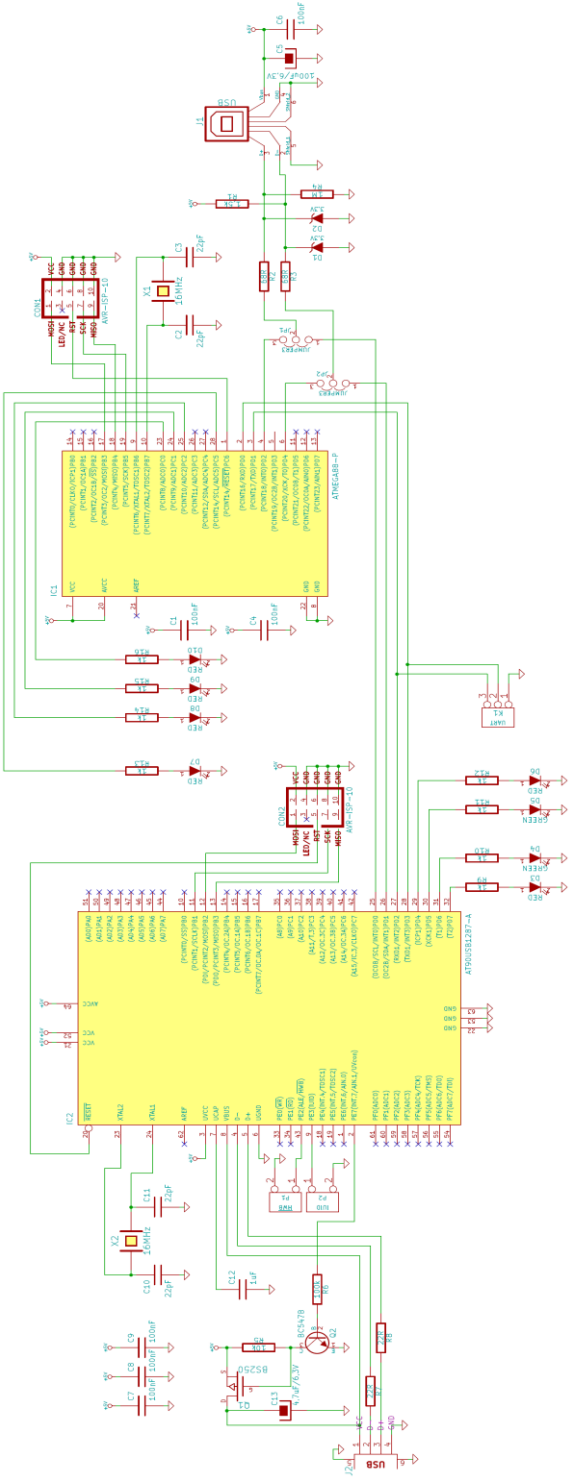

Fig. 1. Scheme of the prototype.

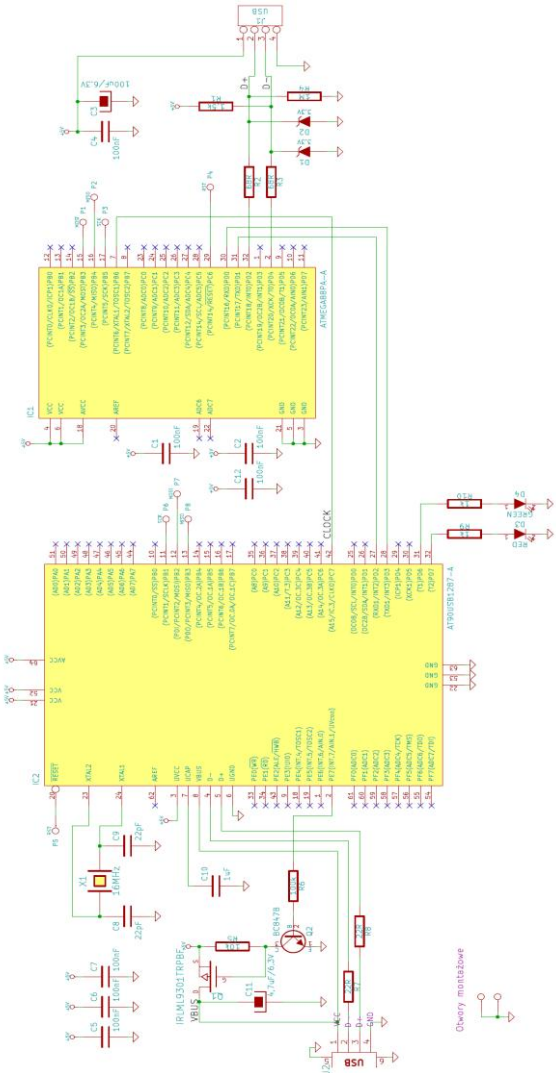

Fig. 2. Scheme of the minimized device.

Fig. 3. Image of the prototype.

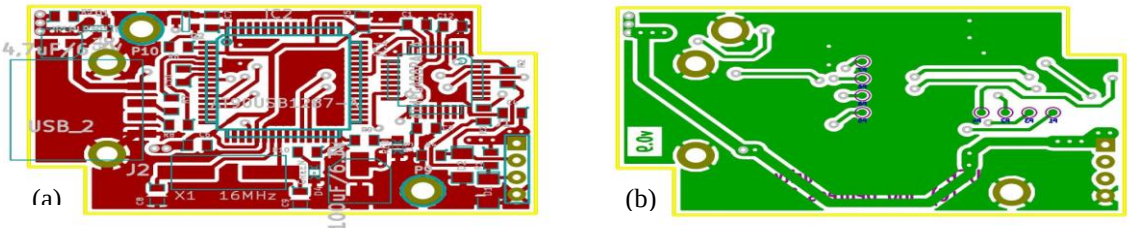

Fig 4. Scheme PCB of miniaturized unit: (a) up layer, (b) down layer in reflection view

Due to this approach, the final device is based on two microcontroller manufactured by *ATMEL*. Using two independent microprocessors connected each other via serial port, allows for easy separation of duties and programming of the device.

The larger microcontroller - *AVR AT90USB1287* [7] is responsible for simulating *USB Host* (supporting HID). It is used for communicating with the keyboard and intercepting data coming out from it. This microcontroller is equipped with hardware support for USB, thanks to that, operating on the port does not take much of the microprocessor time. Due to this fact it may additionally implements the protection algorithms. The microprocessor reads the keyboards descriptor report and next decodes the report into particular components including keystrokes and *Light-Emitting Diode (LED)* states. In case of sending states of LEDs, particular data is inserted into the report and next according to encoded descriptor in to full keyboard report.

Thanks to encoding of the descriptor we can use vast range keyboards. Intercepted keystrokes from the keyboard are transmitted further according to currently used protection algorithm via *Universal Synchronous and Asynchronous Receiver and Transmitter (USART)* protocol to the second microcontroller, which is the smaller *AVR Atmega88* [8]. Incoming data from *USART* port is immediately transmitted to the workstation. This microprocessor does not has hardware support for USB, thus it has to emulate it by software.

In implementation we used two open source libraries – *LUFA* [9] and *V-USB* [10]. The first one helps with programming of *AVR* microprocessors equipped with hardware based USB support. We took an advantage of the available example for *USB Host* from this library. The later one is used for software emulating USB port on the *Atmega88* microcontroller.

While designing the prototype we followed the V-USB authors recommendations related to proper connecting of the USB port to the *Atmega88*. We present the electronic scheme of the prototype in the Fig. 1 and the constructed unit in the Fig. 3. Apart from microcontrollers and USB ports, in the prototype we build in two *In-System Programming (ISP)* ports (to program microprocessors) and LEDs to signalize states of communication.

B. Minimizing the unit

We applied *Surface-Mount Technology* (*SMT*), what let us significantly reduce electronics of the device. We use microprocessors in TSQP covers and most of the larger passive components in covers 0806. Thanks to possibility to make clock signal externally available in microprocessor *AT90USB1287* we resign from second quartz resonator and clock the microprocessor *Atmega88PA* with it. Minimization required necessity of using two-layers PCB, but simultaneously allowed to reduce the circuit board size to 54 x 31mm. Image of the PCB is presented in Fig. 4. Finally, the unit is enclosed within box of external size 60 x 37 x 13mm. The electronic scheme of miniaturized version does not differ much from the prototype and can be found in Fig. 2. Constructed *Surface Mounted Devices (SMD)* device is presented in Fig. 5.

C. Technical data of device

Power supply:
- USB port of workstation
- power consumption ~ 90mA (keyboard not included)

Device is completely compatible with Device Class Definition for Human Interface Devices (HID) and works in boot mode as well.

D. Protection algorithms

As stated earlier we implemented the protection algorithms from [5] on the *AVR AT90USB1287* microcontroller. We will only describe the implementation of the basic algorithm. The idea behind this algorithm is simple, the keystrokes from the keyboard are outputted with a given delay. To do so, we use the microcontrollers timer and a *First In, First Out (FIFO)* list. The keystrokes from the keyboard are stored on the list. Simultaneously, if the microcontroller receives a timer interrupt, it outputs the top element on the list. The other algorithms from [5] are based on a similar idea. There is only a difference in the initialization of the timer. Thus, we omit their description.

Fig. 5. Image of the minimized device.

III. EFFICIENCY TEST AND RESULTS

In this section we present information about the performed tests of the designed solution. To emphasize the differences between normal data coming from the keyboard and data modified by our device (set to threshold time - 100ms) we will show two histograms. First, we used a common text to receive similar digraphs for both tests. Then, we used our software

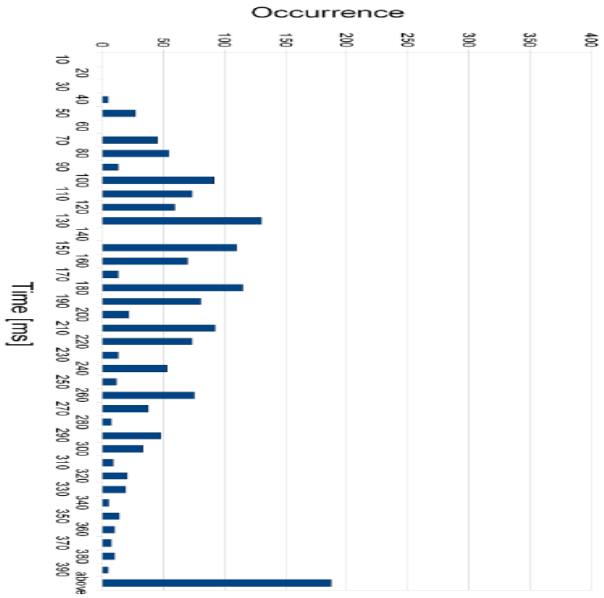

Fig 6. Digraph histogram (normal keystrokes)

– OBiometric to extract timings of digraph using raw data. Finally, we created two histograms (with interval length of 10ms) representing the number of digraphs in the given intervals (see Fig. 6, Fig. 7). The results on Fig. 7 may be confusing, because of some gaps in the histogram. This is implied by fact that our device is not competently accurate i.e. there are some delays in data flow between microcontrollers. This problem could be overcome by using one microprocessor or two more powerful ones (considering implementation security algorithm on the microcontroller witch emulates keyboard).

IV. CONCLUSION

It is important to mention that our hardware-based solution is resistant to any remote attack. It guarantees security at the same time being complete transplant for the user.

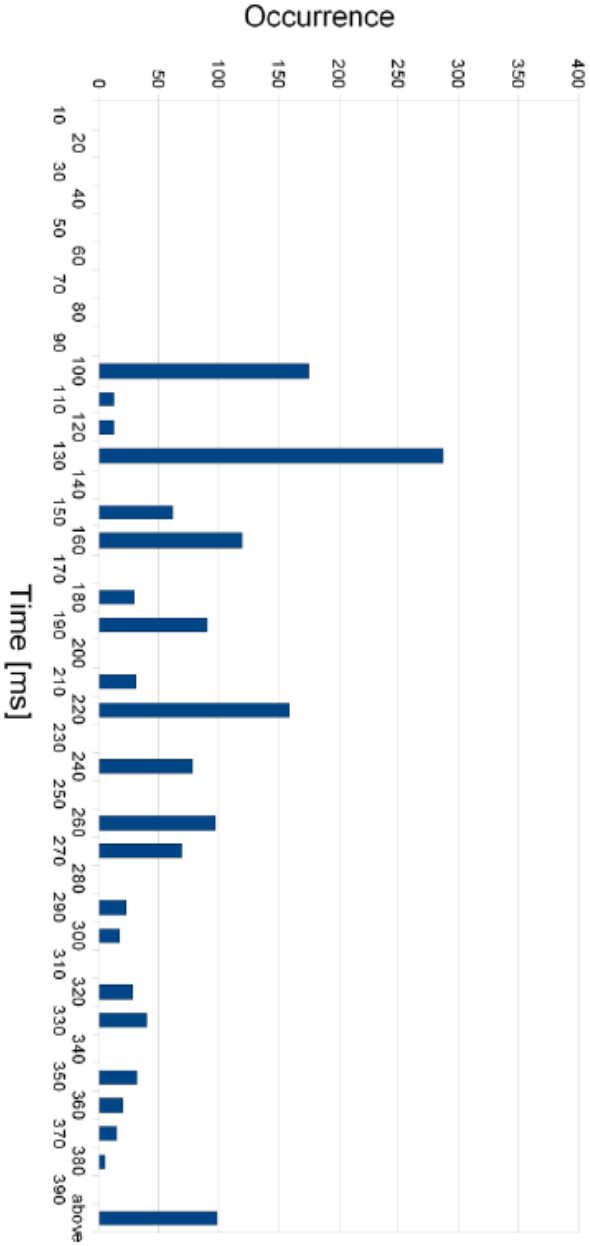

Fig 7. Digraph histogram (keystrokes modified by our device, set to 100 ms)

As a future application for our solution we anticipate incorporation into keyboard's firmware as well as other mobile devices equipped by input interface e.g. virtual keyboard (on touch screen) or pin pad (automated teller machine).

We consider constructing more complex unit, which will be able to perform like biometric authorization system. It means that user should first authorized itself on the unit and later be able to get access to terminal or other resources. Finally the described solution is submitted for patent protection

ACKNOWLEDGMENT

This project was supported by European Union as part of the European Social Fund 8.2.1 - Dolnoslaski Bon na Innowacje.

REFERENCES

1. Giot, R., El-Abed, M., Hemery, B., and Rosenberger, C. (2011). Unconstrained Keystroke Dynamics Authentication with Shared Secret. In Computers & Security 30(6-7), pages 427-445.
2. Fridman, A.; Stolerman, A.; Acharya, S.; Brennan, P.; Juola, P.; Greenstadt, R., and Kam, M., (2013). Decision Fusion for Multi-Modal Active Authentication. In IT Professional 15(4), pages 29-33.
3. Zhong, Y., Deng, Y., and Jain, A. K. (2012). Keystroke dynamics for user authentication. In CVPR Workshops, pages 117–123.
4. Klonowski, M., Syga, P., and Wodo, W. (2012). Some remarks on keystroke dynamics - global surveillance, retrieving information and simple countermeasures. In SECRYPT, pages 296–301.
5. Hanzlik, L., and Wodo, W., (2013). Identity Security in Biometric Systems Based on Keystroking. In SECRYPT, pages 524–530.
6. Universal Serial Bus (USB), Device Class Definition for Human Interface Devices (HID), 2001, [Online]
7. http://www.usb.org/developers/devclass_docs/HID1_11.pdf
8. Documentation for 8-bit Atmel Microcontroller with 64/128 Kbytes of ISP Flash and USB Controller
9. AT90USB646,AT90USB647,AT90USB1286,AT90USB1287, 2012, [Online] http://www.atmel.com/Images/doc7593.pdf
10. Documentation for 8-bit Atmel Microcontroller with 4/8/16K Bytes In-System Programmable Flash ATmega48/V, ATmega88/V, ATmega168/V, 2011, [Online]
11. http://www.atmel.com/images/doc2545.pdf
12. LUFA Library Documentation, 2013, [Online]
13. http://www.fourwalledcubicle.com/files/LUFA/Doc/130303/html
14. V-USB, A Firmware-Only USB Driver for the AVR, [Online]
15. http://vusb.wikidot.com

AUTHORS PROFILE

Wojciech Wodo, MSc is a PhD candidate of computer science at Wroclaw University of Technology, graduate of special Top 500 Innovators program at UC Berkeley focused on science management, technology transfer, commercialization and university-industry collaboration. He received MSc degree from Wroclaw University of Technology in computer science. Mr. Wodo worked as a technology transfer specialist in Wroclaw Research Center EIT+ (2011-2012).His research fields: cryptography, computer security, biometrics.

Lucjan Hanzlik, MSc is a PhD candidate of computer science at Wroclaw University of Technology. Laureate of prestigious program of Foundation for Polish Science – Ventures devoted to digital cryptographic signatures on smart cards. Author of dozen papers about cryptography and computer security. He received MSc degree from Wroclaw University of Technology in computer science. His research field: cryptography, computer security, smart cards.

Konrad Zawada, BSc is a MSc student of electronics at Wroclaw University of Technology. Integrated circuits designer and digital signals specialist, 8-bit microprocessors developer. He received BSc degree from Wroclaw University of Technology in electronics. His research field: integrated circuits, microprocessors, electronics.