ISSN 2300-5149

PRZEGLĄD TELEINFORMATYCZNY

Dawniej: Biuletyn Instytutu Automatyki i Robotyki WAT
ISSN 1427-3578

TOM 2 (20) **Nr 3-4 (38)**
WARSZAWA **2014**

PRZEGLĄD TELEINFORMATYCZNY
TELEINFORMATICS REVIEW

Dawniej: BIULETYN INSTYTUTU AUTOMATYKI I ROBOTYKI WAT
(ISSN 1427-3578)
Ukazuje się od 1995 r.

ADRES REDAKCJI
Redakcja Przeglądu Teleinformatycznego
00-908 Warszawa 49, ul. Gen. S. Kaliskiego 2
tel. 261 83 95 52, fax. 261 83 71 44
e-mail: pt [at] ita.wat.edu.pl
WWW: **http://przeglad.ita.wat.edu.pl/**

Wersją pierwotną czasopisma jest wersja elektroniczna

REDAKTOR NACZELNY:
Antoni Donigiewicz

REDAKTOR WYDANIA
Antoni Donigiewicz

OPRACOWANIE STYLISTYCZNE
Renata Borkowska

PROJEKT OKŁADKI
Barbara Chruszczyk

WYDAWCA: Instytut Teleinformatyki i Automatyki WAT

ISSN 2300-5149
ISSN 2353-9836 (on-line)

Electronic safe for passwords storage

Kamil ŚMIESZEK[1], Janusz FURTAK[2]

[1] Institut of Teleinformation and Authomatics,
Military University of Technology
Gen. S. Kaliski 2 St., 00-908 Warsaw, Poland
kamilsmieszek@gmail.com

[2] Institut of Teleinformation and Authomatics,
Military University of Technology
Gen. S. Kaliski 2 St., 00-908 Warsaw, Poland
jfurtak@wat.edu.pl

ABSTRACT: This paper considers the problem of storing user's sensitive data in a safe manner. Those data include for example: user account names and passwords for any websites, mailboxes, bank account numbers, PIN and PUK codes. This had been solved by creating an application, which allows the user to create his own safe for passwords. The sensitive data are protected by the computer's TPM security features, including an asymmetric encryption algorithms, Root of Trust mechanisms and binding the encrypted data to a specific platform.

KEYWORDS: sensitive data protection, Trusted Platform Module, asymmetric encryption

1. Introduction

Almost in every piece of life the human being has to deal with many services, where the authentication is necessary. This is required for example to access a e-mail server, bank accounts, cloud computing accounts, social websites and so on. The access without giving a login and password is completely impossible. It follows therefore that the human's brain has to remember many authentication data. It is not difficult when the amount of data is small. The serious problem arises when amount of the data is very high. It would be much more convenient for us to use a software tool which supports secure storing and making available the account data.

This paper considers a solution of the problem using an application that allows to create by an user the Electronic Safe for Passwords Storage (ESPS). The safe (containing data of one user) could to be stored on removable storage

media and secured before an unauthorized access. It is easy to use, portable, quite cheap and does not create issues in cooperation with the popular computer operating systems. Those requirements are met by removable storage on Flash RAM. Because continuously deleting and modifying data stored on this kind of memory does not delete the "old" data, but marks the storage area as empty, it possible to read those data using the external software tools. For this reason the data stored in ESPS on a removable Flash RAM drive are always encrypted. In case of storing passwords or the other secrets there, the management application uses strong (dependable) authorization methods which allows users to receive the sensitive data from ESPS. For this purpose the ESPS uses a Trusted Platform Module.

The application consists of two components. The first one is a database, in which the sensitive data are stored in an encrypted form (an AES symmetric encryption algorithm is used). Keys and the other necessary data (for example, database user's authentication data) are stored in Root of Trust supported by TPM (it is the second component). The default place for storing of the database and the Root of Trust is the hard drive, but user of ESPS can also change this place pointing a Flash RAM memory when using of ESPS is needed on the other machine.

The market of password-managing software is rather rich. An important issue is the level of security of such the sensitive data. The obvious thing is passwords and any other sensitive data are stored in databases encrypted and unable to read by an unauthorized user. Different security methods are used. The methods do not allow too easy to guess passwords in case of intercept the database – generally functions implementing encrypting and hashing are used. Examples of implementations of a passwords' safes are described below. These include:

- KeePass,
- LastPass,
- GNOME Keyring,
- Windows Credential Manager,
- OneKey Pro.

KeePass

One of the most popular free, open-source developed software for managing user passwords is KeePass. This application gained its popularity not only by free distribution, but also by a multiplatform design (KeePass is available for: Microsoft Windows, Apple Mac OS X, Linux, Google Android, Apple iOS, Microsoft Windows Phone 7, RIM BlackBerry OS and J2ME), multilingual GUI, possibility of extending features by plugins and, what is most important, a constant and active development. The way that KeePass stores the entire sensitive data in a one tiny little database file. The file can be protected by:

- a Master password;
- a file with key (generated during creation of the database);
- a Windows local account authentication.

All of those things can obviously be applied together, but user can choose between options and decide for the one, that provides a convenient and strong passwords protection.

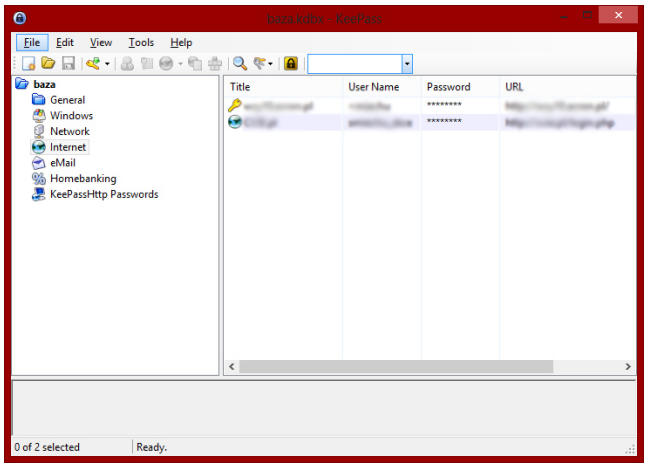


Fig. 1. Screenshot from the KeePass 2

A database file is encrypted by one of the two algorithms (i.e. AES or Twofish). Passwords stored in are also hashed by an SHA-256 hash function. All the encryption mechanisms protect the sensitive data from the dictionary and brute-force attacks, but only in a some way. User password is hashed by SHA-256 and then the result is N-times encrypted by an AES algorithm. Finally, there is generated for the cryptogram a SHA-256 hash again. All of that cannot be considered as 100% certain security solution, but make a very serious obstacle for a potential attacker. It should be taken into consideration that not only passwords are stored in a database but also the other user's information, such as account name or user personal notes and protected too. Even during the normal work all of the data are loaded to the computer's RAM memory encrypted.

LastPass

The similar but a bit different functioning tool is LastPass. This password manager almost fully depends on the cloud computing solutions, so users can access their sensitive data using only an Internet browser with the appropriate plugins or mobile applications. Very basic advantage of that solution is

a synchronization passwords between much variety of devices, that is very easy. The security mechanisms of LastPass are:

- a master password;
- a key file;
- a two-step verification with Google Authenticator or (hardware managed) with a Yubikey.

The idea of storing passwords is very different from a KeePass, but the encryption is rather similar. There are also used AES and SHA-256 with salt algorithms.

GNOME Keyring

GNOME Keyring is a collection of components in a Unix/Linux GNOME environment that is able to store passwords, keys or certificates and make them available to user's applications. GNOME Keyring is in fact an OS user session deamon which combines the stored classified data with a logged-in user session. Every password is encrypted by the encrypting-hashing pair – AES-SHA256. The master password for the secured passwords storage is the same as the one that user requires to log in. GNOME Keyring is a part of the Linux PKCS#11 infrastructure, what makes possible to store storage keys on the smart cards.

Windows Credential Manager

Like a GNOME Keyring in Linux OS, there is also a password manager in Microsoft Windows OS family (available in Windows 2000 and versions above). Windows Credential Manager allows for managing websites access and network access data. The security is close to the Unix/Linux solutions – the main password is the same password as the user logs in the OS.

Credential Manager uses DPAPI (Data Protection API) to encrypt data. The key for them (MasterKey) is encrypted based on PKCS#5 standard, which means generating key directly from the master password. This combined with the 3DES symmetric algorithm decides about the safety of the key stored in a user profile default folder.

For all of the application that were described above, one common issue can be notice: they are not benefit from the Trusted Platform Module features. For this reason, using the TPM means of encryption will be applied to the Electronic Safe for Passwords Storage application.

2. Concept of solution

2.1. The idea of storing sensitive data in a secure way

The place to store user's ID's and passwords is a relational SQLite database. This one that is being created for the electronic safe for passwords storage application purposes is simple (the database structure is shown on Fig. 2) and consists of two tables:

1. **tblCategories** – the table for storing passwords categories, into which an user can separate his sensitive data. Columns included in this table are:
   a) ID (data type: INTEGER) – category ID, primary key of the tblCategories table;
   b) Name (data type: BLOB) – category name (encrypted) (i.e. mobile phone, e-mail address, bank account);
   c) ParentID (data type: BLOB) – category parent ID (encrypted).
2. **tblPasswords** – the table for storing user's sensitive data, i.e. account names and passwords. Columns included in this table are:
   a) ID (data type: INTEGER) – classified data record ID, primary key of the tblPasswords table;
   b) authLogin (data type: BLOB) – user name/bank account number (encrypted);
   c) authPasswd (data type: BLOB) – password/PIN code (encrypted);
   d) catID (typ danych: BLOB) – category ID (encrypted), to which belongs encrypted password record, a foreign key connected to a tblCategories table.

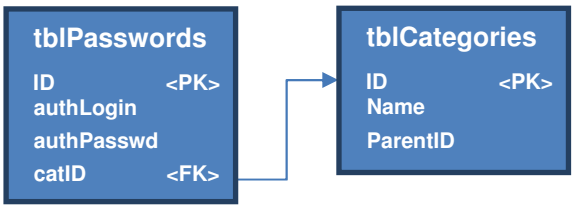


Fig. 2. Safe for passwords storage – database structure

Selected database management system does not provide strong security mechanisms for stored data. The unsecured database can be easy broken by simple SQL statements, what allows the potential attacker to have unauthorized access to sensitive data. The ESPS application has to protect those data in much more stronger way.

In the database can be stored data belonging to multiple users. All fields of database records belonging to a certain user (indicated in the above description as "encrypted") are saved in the database in encrypted form using a symmetric algorithm AES-256 and a key of the user (the key of user is stored in Root of Trust supported by TPM).

AES-256 is part of the implementation of an ESPS. Any attempt to access data from the database after reading the record from database needs to decrypt it using a correct key.

2.2. Idea of storing the keys

The keys necessary for reading the data from the database are stored in a special structure, which guarantees a safe storage of these keys. The keys necessary for reading the data from the database are stored in a special structure, which guarantees a safe storage of these keys. This structure is hierarchical and is called Root of Trust. Creating and using this structure is supported by the Trusted Platform Module (TPM)[1].

On top of this hierarchy is asymmetrical Endorsement Key (EK). This key is generated once for each TPM (is not possible re-generation of the key). EK is stored in the TPM resources. The private portion of this key is not available outside of the TPM and is used to verify the signature Storage Root Key (SRK). Asymmetrical SRK is created during the procedure of taking over ownership of the TPM. Its private part is signed by public part of EK. SRK is also stored in the TPM resources. SRK is the parent of each key generated for individual users of ESPS. Sample structure Root of Trust, is shown in Fig. 3.

For each user of ESPS is necessary to generate the following set of cryptographic keys:

- **RK** – symmetric key and initialization vector which are used for AES encrypting of records in database belonging to the individual user,
- **DBK** – symmetric key and initialization vector which are used for AES encrypting of the database file that contains only the user's data – the file can be stored for instance in Flash memory,
- **sKey1** and **sKey2** – asymmetric keys for signing keys respectively **DBK** and **RK**,
- **STK** – asymmetric key for signing keys of individual user.

[1] TPM is an implementation of a standard developed by the Trusted Computing Group [6]. This module is designed to support the cryptographic procedures and protocols that can be used for securing data [7]. Trusted Platform Module provides the following functions: generating an asymmetric key pair, secure storage of keys, generating an electronic signatures, encryption and decryption and implementation of an operation defined by the standard PKCS #11.

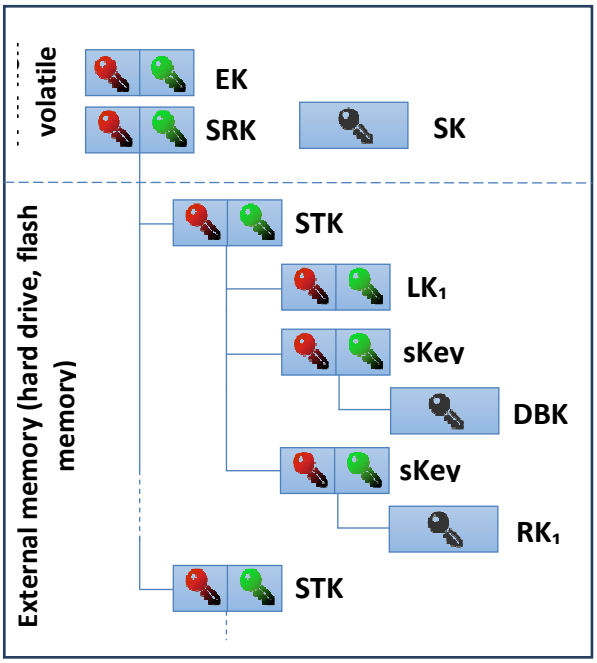


Fig. 3. Sample structure Root of Trust

Once the database is closed, but before closing the ESPS application, database file that contains all user data, is encrypted using the AES-256 algorithm and key SK, SK is owned by TPM owner and is stored in a secure NVRAM of TPM.

2.3. The concept of solution and application architecture

Application of the ESPS consists two subsystems [8]:

1. Subsystem for TPM key management, including a `.DLL` dynamic library used for TPM operations,
2. Subsystem for user's sensitive data management, including:
   a) application's GUI,
   b) `.DLL` library used for encrypting and decrypting data and database file with an AES algorithm,
   c) `.DLL` library to manage the database.

In order to provide full functionality all of those components must be stored in the same folder together with the Application executable file (on a hard drive or Flash memory).

In the implementation of ESPS are used the following software components:

- graphical interface,
- library functions to encrypt and decrypt data,
- library functions to handle SQLite database,
- library functions to handle the TPM.

Implementation of presentation layer of electronic safe uses the Windows Forms interface with .NET Framework 3.5 [3] This part includes:

- TPM management (taking ownership, erasing ownership data, checking TPM state, etc.),
- operations on categories of the sensitive data management (adding, showing, modifying, deleting),
- sensitive data management (adding, showing, modifying, deleting),
- generating the TPM key hierarchy,
- erasing the TPM key hierarchy,
- saving the TPM key hierarchy and the database file on a Flash memory drive.

The encryption-decryption `.DLL` library uses an `Aes` from the .NET Framework `System.Security. Cryptography` namespace. This includes:

- encrypting the text or the byte stream with the AES-256 algorithm in CBC encryption mode using the IV vector and a key provided by the TPM,
- decrypting encrypted data using given IV vector and a key provided by the TPM.

The `.DLL` library that is used to manage the SQLite database uses the ADO.NET `System.Data.SQLite` adapter. It realizes the following jobs:

- executing SQL queries,
- adding SQL queries parameters,
- creation of the database,
- SQL database connection management,
- SQL database transaction management.

The last `.DLL` library is the one that directs tasks of the TPM module. It was developed using the TSS standards (*Trusted Computing Group Software Stack*) [2]. The TSS library together with TPM module and `System.Security. Cryptography.Aes` class is the fully functional

encryption subsystem which is used in the implementation of the electronic safe. The TSS was developed by IBM company and standardized by TCG. It is a specification that makes creating the TPM applications easier. Functions of the TSS are:

- sending and receiving low-level commands,
- sending and receiving data streams,
- the key storage management (those are being stored outside the TPM),
- registering and management of the occurred events.

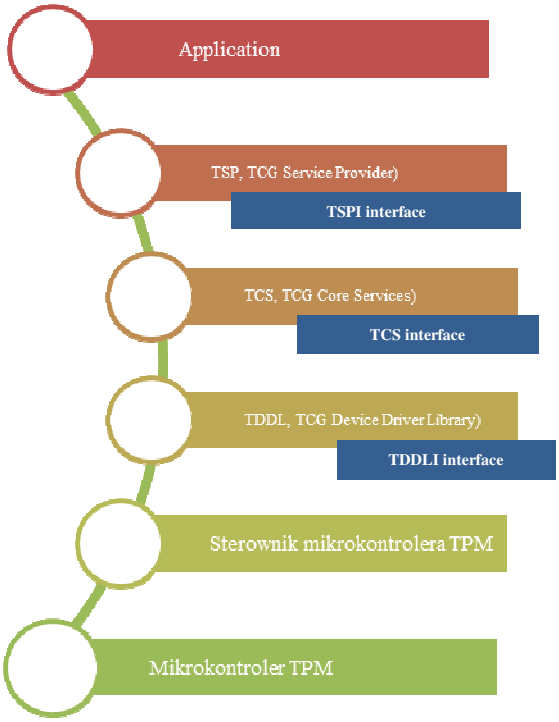


Fig. 4. The TSS structure

To improve functionality the TSS library was divided into separate layers connected with clearly defined interfaces. To provide a communication between the application and TPM, the data has to go through all of those layers. AT first the application communicates with the TSS service provider (TSPI interface). The TSPI-connected TCS layer manages the TPM resources such as session authorization or conversion of the TCS commands to byte streams understandable for the TPM. The TDDL library then decides about a communication with the TPM driver offering a small piece of instructions including opening and closing access to the TPM driver, sending and receiving

data to/from it and requesting the driver about its properties and cancelling commands sent to TPM by the upper layers [4]. Communication with TPM is implemented using the TSS implementation, TrouSerS for Windows (0.3.6 version) [5].

2.4. Using of ESPS

The application of ESPS is a software to protect user's sensitive data. The first operations you must make are as follows: taking ownership of the TPM and setting a password for access to the DBK key stored in resources of TPM. The process of taking ownership of the TPM is done using the ESPS. The user is asked to enter a password which is required during operations: resetting the TPM and generation of Root of Trust. In the second step the necessary hierarchy of asymmetric keys is generated. This process involves creating a Root of Trust and the database in which user's sensitive data will be stored. During this operation on your hard drive (or flash storage media) in the directory containing the application executable file are saved the following files:

- an empty database with structure ready for storing data, `PasswordSafeDB.db`
- DBK.key for **DBK** key,
- RK.key for **RK** key.

Without those files written on a hard drive or Flash memory, the management of user's classified data is impossible. If files are created successfully, the proper menu fields in the application will be activated, which is shown on Fig. 5.

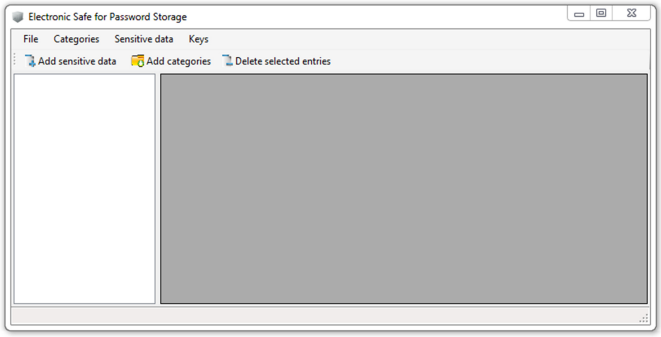


Fig. 5. The screen view of ESPS after successful creation of all the files

User can insert his passwords or PIN codes to the previously created database just after adding at least one data category. Those categories can be a

primary or a secondary. The difference is, that the primary categories hasn't got another parent primary category, but the primary category is a parent for secondary level category. The secondary category is connected to only one primary category. This structure a bit reminds a structure of file system in OS. Entered data will be shown in the main application screen assigned to chosen category (Fig. 6).

Double-click on chosen line in the right part of the window opens a next window that allows you to copy and paste data into another application i.e. on a website. An example of such a window is shown in Fig. 7.

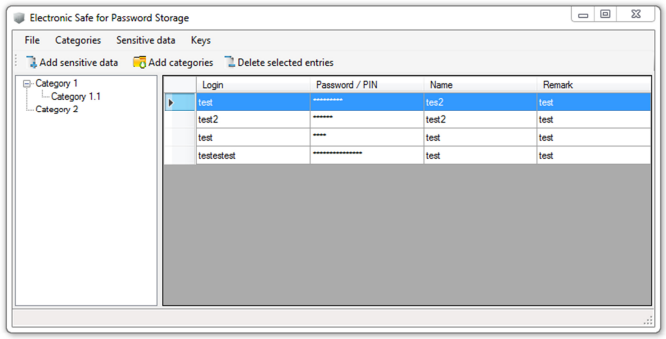


Fig. 6. ESPS after adding three categories and four records of sensitive data

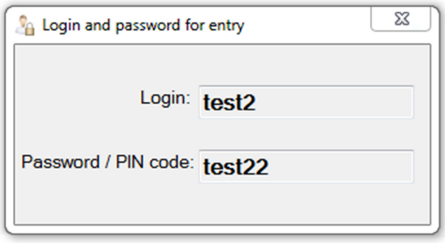


Fig. 7. An example of a window to copy and paste data

The database with user's sensitive data is stored in the same folder, together with the ESPS application executable file. User can also store his data on external Flash memory drive. These data are just the copies of all the original application files (**PasswordSafeDB.db, DBK.key** and **RK.key**). To improve data security, files are also added to a ZIP archive and protected by a password, which is required to create an archive file. ESPS can read the content of this archive itself – user doesn't need to extract it manually.

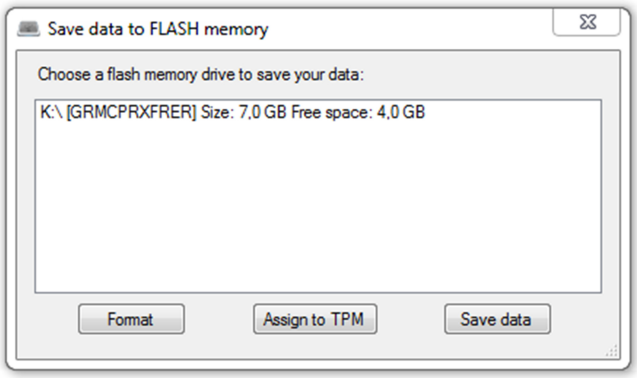


Fig. 8. A window that allows to create a ZIP archive file

3. Conclusion

Using a Trusted Platform Module to protect user's sensitive data gives a significant increase protection of these data. The potential attacker cannot get access to them, if he doesn't know the password, even in case of stealing a computer. This is possible, because of one of the TPM features i.e. a special counter for unsuccessful authentications [1]. If the factory-implemented counter value is exceeded, a TPM locks itself before access from outside for some time, what is significant inconvenience for the attacker. The obvious problem of the latest safe's implementation is lack of possibility to export the whole key structure to and import them from TPM of another computer. For this reason it is not possible to recover protected data except the environment of computer with the original TPM. It is very important to improve the application in that aspect, what will be the first job to do in later implementations.

References

[1] *TPM Main Part 1 Design Principles. Specification Version 1.2. Revision 116*, Trusted Computing Group, Incorporated, 2011.

[2] *TCG Software Stack (TSS) Specification Version 1.2 Part1: Commands and Structures* (http://www.trustedcomputinggroup.org /files/resource_files/6479CD77-1D09-3519-AD89EAD1BC8C97F0 /TSS_1_2_Errata_A-final.pdf).

[3] MAYO J., *C# 3.0 dla .NET 3.5. Księga eksperta*, Helion, Gliwice, 2010.

[4] *Trusted Platform Module Library. Part 1: Architecture* (http://www.trustedcomputinggroup.org/files/static_page_files/7F7F6AFE-1A4B-B294-D0EE43535A6176B2/TPM%20Rev%202.0%20Part%201%20-%20Architecture%2000.96%20130315.pdf),

[5] CHALLANER D., YODER K., CATHERMAN R., SAFFORD D., VAN DOORN L., *A Practical Guide to Trusted Computing*, IBM Press, Boston, 2007.

[6] *TPM Main Part 1 Design Principles. Specification Version 1.2. Revision 116*, Trusted Computing Group, Incorporated, 2011.

[7] KINNEY S., *Trusted platform module basics: using TPM in embedded systems,* Embedded Technology Series, Elsevier Inc., Amsterdam, 2006.

[8] VIEGA J., MESSIER M., C i C++. *Bezpieczne programowanie. Receptury*, Helion, Gliwice, 2005.

Elektroniczny sejf do przechowywania haseł

STRESZCZENIE: Artykuł dotyczy problemu bezpiecznego przechowywania wrażliwych danych użytkownika. Te dane mogą obejmować: nazwy kont i treść haseł do aplikacji internetowych lub do skrytek poczty elektronicznej, numery kont bankowych, numery PIN do kart kredytowych, numery PIN i PUK do telefonów komórkowych. Zaproponowane rozwiązanie pozwala użytkownikowi na utworzenie jego własnego sejfu na te dane. Dane w tym sejfie będą zabezpieczone kryptograficznie. Do wspomagania tych zabezpieczeń jest wykorzystany moduł Trusted Platform Module, a w szczególności do tworzenia kluczy asymetrycznych, do zbudowania i utrzymywania drzewa zaufania dla aplikacji, szyfrowania/odszyfrowywania wrażliwych danych użytkownika oraz do uwierzytelniania użytkowników sejfu.

SŁOWA KLUCZOWE: zabezpieczanie wrażliwych danych, Trusted Platform Module, szyfrowanie niesymetryczne.

Received: 25.11.2014

Przegląd modeli zaufania PKI

Piotr OLSZEWSKI

Wydział Cybernetyki, Wojskowa Akademia Techniczna,
ul. Gen S. Kaliskiego 2, 00-908 Warszawa
polszewski@wat.edu.pl

STRESZCZENIE: Infrastrukturę klucza publicznego można opisać jako zbiór technologii, urządzeń, polityk, procedur oraz ludzi, który umożliwia zarządzanie certyfikatami klucza publicznego oraz zaufaniem dwóch lub więcej stron transakcji do siebie. W tym celu wprowadzono struktury nazywane modelami zaufania. Modele te można podzielić na kilka grup, w zależności od cech wspólnych, w szczególności podobnej struktury oraz sposobu zarządzania relacjami pomiędzy poszczególnymi węzłami w strukturze. Każda grupa posiada cechy wspólne z innymi oraz część, która została zmodyfikowana w porównaniu z dwoma podstawowymi modelami: hierarchicznym oraz rozproszonym. Celem niniejszej publikacji jest przegląd modeli zaufania, prezentacja ich cech charakterystycznych oraz krótka ocena ich zalet i wad w praktycznym zastosowaniu.

SŁOWA KLUCZOWE: PKI, zaufanie, modele zaufania, X.509

1. Wprowadzenie

Infrastruktura klucza publicznego – PKI (ang. *Public Key Infrastructure*) jest to zbiór urządzeń, oprogramowania, polityk bezpieczeństwa, procedur oraz użytkowników, umożliwiający tworzenie, przechowywanie, zarządzanie i dystrybucję certyfikatów klucza publicznego, stanowiących podstawę do identyfikowania tożsamości ludzi lub maszyn w sieciach komputerowych (np. Internet). Zgodnie z polską normą PN-I-2000 certyfikat klucza publicznego jest taką informacją o kluczu publicznym danego podmiotu, która dzięki podpisaniu przez zaufaną trzecią stronę jest niemożliwa do podrobienia [1]. W odniesieniu do powyższego, kluczową sprawą zarówno certyfikatu klucza publicznego, jak i całej infrastruktury klucza publicznego (jako że korzysta ona z certyfikatów) jest zaufanie oraz możliwości określenia jego poziomu (a w szczególności jego braku).

Podstawową kwestią jest zdefiniowanie samego terminu zaufania. W literaturze, w zależności od źródeł, funkcjonuje wiele definicji tego pojęcia. Zgodnie z obowiązującymi normami, zaufanie to przekonanie o niezawodności i prawdziwości informacji lub zdolności danego podmiotu do zachowania w sposób właściwy w określonym kontekście [2]. Definicja ta zastąpiła dotychczas stosowaną, znajdującą się uprzednio w normach ITU-T X.509 (2008 wraz z późniejszymi erratami). W bardziej ogólnym ujęciu, zaufanie można określić jako coś, co jest niezbędne dla kanału komunikacyjnego, ale nie może być przesłane od źródła do celu za pomocą tego kanału [3]. Na podstawie ogólnej definicji, idąc przez definicje pochodne Gercka, można dokonać próby sformułowania definicji technicznej: *Zaufanie jest miarą tego, co oszacował obserwator A na temat zachowania podmiotu B w czasie T dla przypadku X*. Z powyższej definicji wynika, że powinien istnieć sposób, w którym jedna strona ma szansę nabycia przekonania, czyli dokonania oszacowania, o tym, że druga strona zachowa się we właściwy sposób.

Struktura pozwalająca określić poziom zaufania w obszarze działania PKI nazywana jest modelem zaufania. Do jego podstawowych zadań należy określenie, którym certyfikatom jednostka może zaufać, dostarczenie sposobów tworzenia związków zaufania pomiędzy różnymi jednostkami oraz udostępnienie metodyki ograniczania i kontrolowania zaufania w zadanym środowisku [4].

Aktualnie w literaturze spotkać można różne koncepcje modeli zaufania. Niektóre z nich są wdrażane w rzeczywistych systemach, inne z kolei pozostają w sferze rozważań teoretycznych i nie wchodzą do codziennego użytku. Celem części właściwej niniejszej publikacji jest próba przeglądu dostępnych struktur modeli zaufania, przedstawienie ich charakterystyki oraz próba oceny ich użyteczności w rzeczywistych przypadkach, w jakich stosuje się PKI do zarządzania tożsamością użytkowników. Temat wyboru modelu zaufania jest istotny z punktu widzenia każdego podmiotu korzystającego z PKI, ponieważ z jednej strony należy uwzględniać rachunek ekonomiczny takiego wyboru (koszty związane z infrastrukturą, utrzymaniem jej itp.), z drugiej zwracać uwagę na bezpieczeństwo oraz użyteczność oferowaną przez konkretny model. Ze względu na fakt, iż z systemów kontroli tożsamości korzystają dziś nie tylko firmy i organizacje, ale również instytucje państwowe, urzędy, służby, temat doboru odpowiedniego modelu jest zawsze aktualny.

2. Modele zaufania

Modele zaufania pozwalają definiować związki zaufania pomiędzy poszczególnymi elementami infrastruktury. W ujęciu formalnym modele zaufania przedstawiane są jako grafy nieskierowane, w niektórych przypadkach

(modele z ustaloną hierarchią, brak certyfikacji wzajemnej wewnątrzdomenowej) otrzymuje się szczególny rodzaj grafu, jakim jest drzewo. Poszczególne elementy PKI (organy, podmioty itp.) są węzłami w grafach, natomiast krawędzie opisują relacje, w jakie wchodzą między sobą węzły nimi połączone (relacja certyfikacji, certyfikacji wzajemnej).

Najbardziej popularnym związkiem jest relacja *certyfikacji*, kiedy jeden element (zazwyczaj CA – *organ certyfikacji*) wydaje certyfikat (a zatem poświadcza prawdziwość zawartych w nim informacji) innemu elementowi (innemu CA, użytkownikowi końcowemu, oprogramowaniu, stronie internetowej) [1]. Jednym z istotnych, dodatkowych mechanizmów, pozwalających na tworzenie relacji w modelach zaufania jest certyfikacja wzajemna. Ma ona miejsce, kiedy dwa organy certyfikacyjne nie będące ze sobą bezpośrednio powiązane ustanawiają relację zaufania, podobną do procesu certyfikacji. Technicznie, certyfikacja wzajemna może odbywać się w sposób identyczny jak standardowa certyfikacja, z wyłączeniem faktu, iż obie strony są wtedy organami certyfikacyjnymi. Rozróżnia się certyfikację wewnątrzdomenową, kiedy obydwa CA pochodzą z tej samej domeny (struktury) PKI oraz międzydomenową, w przypadku gdy certyfikacja wzajemna zachodzi pomiędzy CA pochodzącymi z różnych domen. Zazwyczaj proces certyfikacji wzajemnej zachodzi na poziomie nadrzędnych CA, które poprzez wykonanie tego procesu umożliwiają społecznościom należącym do rozdzielnych domen PKI ustanowienie bezpiecznej komunikacji. Pozwala to na zachowanie własności drzewa dla hierarchicznych modeli, ponieważ użycie certyfikacji wzajemnej na niższych stopniach hierarchii może prowadzić do zaburzenia acykliczności.

Istotne, z punktu widzenia modelowania relacji zachodzących pomiędzy poszczególnymi węzłami, jest to, iż certyfikacja wzajemna nie musi być procesem dwukierunkowym, tj. dopuszczalna i w pełni poprawna jest sytuacja, w której jedno CA certyfikuje drugie, ale drugie nie certyfikuje pierwszego. Wykorzystując narzędzia dostarczane wraz z X.509 w wersji trzeciej, możliwe staje się za pomocą certyfikacji wzajemnej włączenie do bezpiecznej komunikacji tylko części społeczności drugiej domeny – kontrola za pomocą ograniczeń (rozszerzeń X.509) pozwala wybierać poszczególne gałęzie przestrzeni nazw, które dane CA ma obsługiwać.

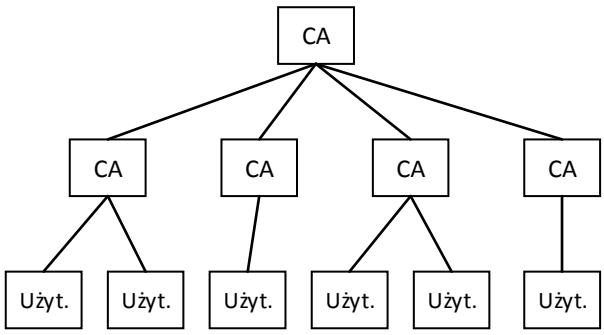
Modele zaufania pozwalają również określić, w jaki sposób należy obchodzić się z certyfikatami, które z różnych powodów muszą zostać unieważnione przed datą ich wygaśnięcia. Dzieje się to najczęściej w momencie, gdy zostanie ujawniony klucz prywatny skojarzony z certyfikatem klucza publicznego danego podmiotu. Do innych możliwości należy zaliczyć np. złamanie wystawcy certyfikatu lub istotną zmianę danych związanych z tożsamością podmiotu, wymuszającą konieczność odwołania dotychczasowego oraz wydanie nowego certyfikatu. Nieodzowne staje się wtedy

poinformowanie możliwie największej liczby użytkowników o tym, że podmiot posługujący się tożsamością może być fałszywy lub nie zawierać aktualnych danych. Automatycznie, węzły wchodzące w skład danego PKI powinny odmawiać uznania unieważnionego certyfikatu. W celu dystrybucji unieważnionych certyfikatów powstały CRL (ang. *Certificate Revocation List*), które są listą certyfikatów, którym już nie można ufać.

2.1. Model ściśle hierarchiczny

Model o strukturze drzewa, w którym wszystkie węzły w hierarchii obdarzają zaufaniem jeden CA. Taki CA nosi nazwę kotwicy zaufania (ang. *trust anchor*). Drzewo może składać się z wielu warstw, reprezentujących pośrednie CA, zaś na końcu drzewa znajdują się użytkownicy. Szczególnym przypadkiem modelu ściśle hierarchicznego jest scenariusz, gdy od razu po korzeniu (głównym CA) występować będą odbiorcy certyfikatów, czyli użytkownicy danego PKI. Scenariusz taki jest niekiedy rozróżniany jako oddzielny model i nazywany modelem jednego CA (z ang. *Single-CA model*) [5]. W niniejszym artykule zdecydowano się go zaklasyfikować jako szczególny przypadek hierarchicznego modelu, ze względu na identyczność założeń i reguł stosowanych w modelu hierarchicznym. Model o strukturze drzewa jest podstawowym modelem używanym w standardzie X.509 [6].

Każda jednostka w ścisłej hierarchii musi otrzymać kopię certyfikatu głównego CA, która po instalacji stanowi podstawę do przetwarzania certyfikatów dla wszystkich połączeń w modelu. Podstawowy CA stanowi zatem nie tylko początek całego drzewa, ale również jest bazą zaufania dla każdego węzła oraz każdego liścia w drzewie. Przykładowa struktura takiego modelu zaufania przedstawiona została na rysunku 1.



Rys. 1. Model ściśle hierarchiczny

Należy zauważyć, że w przypadku wielowarstwowego drzewa, jednostki końcowe (liście drzewa) są certyfikowane przez jednostki znajdujące się bezpośrednio nad nimi w hierarchii, ale bazą nadal pozostaje dla nich główny CA (czyli kotwica zaufania). Wymusza to konieczność przetworzenia ścieżki (drogi), pozwalającej na potwierdzenie zaufania podczas interakcji dwóch użytkowników.

Celem przetwarzania ścieżki certyfikatu jest odnalezienie drogi prowadzącej od certyfikatu docelowego do zaufanego certyfikatu lub kotwicy zaufania, w przypadku modeli ściśle hierarchicznych. Przetwarzanie ścieżki certyfikatów jest zasadniczo podzielone na dwa etapy:

• budowę ścieżki (*path discovery*), która obejmuje agregację wszystkich certyfikatów, które są potrzebne do budowy ścieżki;

• zatwierdzenie (*walidację*) ścieżki, które obejmuje kontrolę każdego certyfikatu na ścieżce i sprawdzenie, czy zawarte tam informacje są prawidłowe.

Budowa ścieżki jest procesem skomplikowanym, zazwyczaj związanym z wykorzystaniem algorytmów grafowych, w których zadanie polega na odnalezieniu drogi łączącej podmiot weryfikujący tożsamość z urzędem, który wydał certyfikat, który ten podmiot stara się zweryfikować. Proces ten może obejmować budowę ścieżki z wykorzystaniem certyfikacji wzajemnych pomiędzy różnymi CA. Do poprawnego działania przedstawionego modelu konieczna jest stała obecność wszystkich węzłów pomiędzy użytkownikami końcowymi, przede wszystkim jednak stała dostępność kotwicy zaufania. W przypadku fizycznego rozproszenia elementów PKI wprowadza to oczywiście narzut czasowy, związany z uzyskaniem informacji. Jednocześnie wyklucza proste użycie modelu w sieciach mobilnych, gdzie węzły mogą dynamicznie podłączać oraz odłączać się od infrastruktury klucza publicznego.

Podstawowym problemem związanym z modelem ściśle hierarchicznym jest konieczność bezpiecznego dostarczenia kopii klucza publicznego głównego CA. Realizacja takiej czynności przy dużej ilości węzłów w drzewie staje się pracochłonna, przy uwzględnieniu np. fizycznego transportu takiego klucza lub stanowi o konieczności ustanowienia dodatkowego bezpiecznego kanału komunikacji, tylko do celu przesłania zadanego klucza. Zazwyczaj w takich przypadkach CA posiadają bezpieczne połączenie z lokalnymi RA (*organy rejestracyjne*), do których fizycznie zgłasza się użytkownik końcowy po swoją kopię certyfikatu. Rodzi to oczywiście dodatkowy nakład finansowy, gdyż konieczne jest utrzymanie fizycznych struktur zarządzania tożsamością, jak również zmniejsza wygodę użytkowania PKI, ze względu na obowiązek osobistego stawiennictwa w RA.

Przełamanie bezpieczeństwa klucza prywatnego głównego CA skutkuje brakiem zaufania w całej hierarchii, co wymusza silną ochronę miejsca, w którym znajduje się kotwica zaufania, zarówno fizyczną, jak i elektroniczną.

W związku z opisanymi problemami model ten znajduje głównie zastosowanie w niewielkich firmach, które są w stanie w łatwy sposób nadzorować cały proces wydawania i przekazywania zaufanej kopii klucza głównego CA, jak również są w stanie wewnątrz swoich struktur odpowiednio chronić korzeń całego drzewa. Złamanie bezpieczeństwa pośrednich CA nie jest już tak dotkliwe, gdyż wymusza unieważnienie tylko poddrzewa związanego z jednym, konkretnym CA, podczas gdy reszta systemu może nadal działać bez zarzutu. Informacja o unieważnionym CA musi zostać rozpropagowana po całej hierarchii, za pomocą odpowiedniego wpisu w listach unieważnień.

2.2. Luźna hierarchia CA

Koncepcja luźnej hierarchii urzędów certyfikacji jest modyfikacją poprzedniego modelu, w którym wprowadzono możliwość pominięcia głównego CA podczas przetwarzania ścieżki certyfikacji, w przypadku gdy obydwie strony zostały certyfikowane w tym samym CA. Pominąć można nie tylko główne CA, ale również te wszystkie, które są nadrzędne do CA, z którego pochodzą certyfikaty stron. Jeżeli podmioty certyfikowały się w różnych CA, procedura przebiega tak, jak w przypadku modelu ściśle hierarchicznego, a zatem konieczne jest przejście całej ścieżki certyfikacji.

Zaletą luźnej hierarchii urzędów certyfikacji jest przyspieszenie weryfikacji dla stron będących dla siebie lokalnymi. W pozostałych przypadkach model ten odzwierciedla wady i zalety ścisłej hierarchii. Zasadniczo zatem zmianom ulegają jedynie reguły przetwarzania ścieżki certyfikatu.

2.3. Hierarchie oparte na regułach

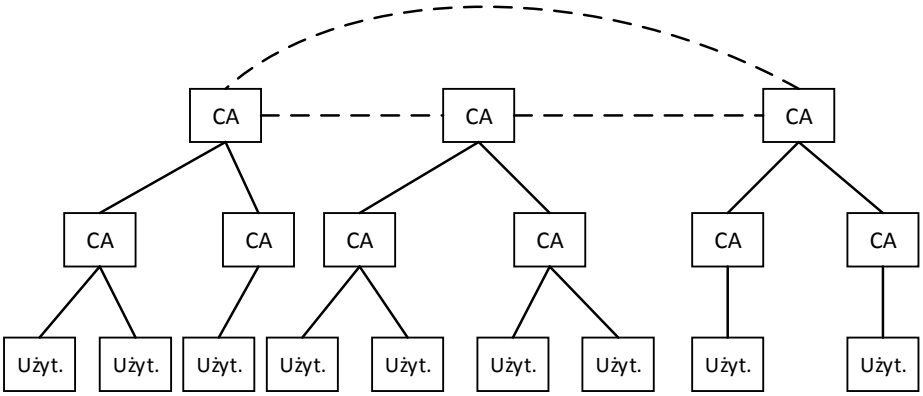
Teoretyczny model zaufania, który rozszerza modele hierarchiczne. Przy tradycyjnym podejściu w poprzednich modelach, podległe CA posiadają tylko jeden podmiot nadrzędny, dzięki czemu podczas tworzenia sieci CA otrzymuje się strukturę drzewa. W ramach takiego drzewa zakłada się, że wszystkie CA działają według wspólnych reguł certyfikacji. W hierarchiach opartych na regułach dodaje się to, że zadany CA może być przypisany do więcej niż jednej reguły, co implikuje, iż może on wtedy należeć do więcej niż jednej hierarchii naraz. Przekłada się to wtedy na fakt, że taki CA mógłby podlegać więcej niż jednemu podstawowemu CA. Oczywiście zakłada się, iż podział na reguły jest podziałem logicznym, przez co należy rozumieć, że możliwa jest również taka sytuacja, gdy istnieje tylko jedno podstawowe CA i to właśnie ono wprowadza

podział w hierarchii na wiele zestawów reguł. Możliwa jest wtedy realizacja modelu wielokrotnych hierarchii [8].

Zaletą takiego modelu jest zwiększona odporność na ujawnienie klucza prywatnego, przy założeniu, że dla każdego zestawu reguł zostanie wygenerowana inna para kluczy. Ze względu na brak znanej, praktycznej implementacji, ciężko jest odnieść się do możliwych wad danego modelu [4].

2.4. Modele rozproszone

Rozproszona architektura opiera się na istnieniu co najmniej dwóch CA, które określają swoje domeny PKI, w ramach których może funkcjonować dowolny model zaufania (np. hierarchia). Każda z takich hierarchii może być płytka, co stwarza architekturę określaną jako w pełni równorzędną, gdyż wszystkie CA są od siebie niezależne. Natomiast w przypadku hierarchii wielopoziomowej wynikiem może być architektura pełnego drzewa – podstawowe CA są ze sobą połączone, ale każdy z nich jest nadrzędny względem jednego lub więcej podrzędnych CA. Możliwe jest również połączenie hybrydowe, gdzie występują zarówno płytkie, jak i wielopoziomowe hierarchie poniżej podstawowych CA. Rysunek 2 przedstawia ogólny model rozproszonej architektury zaufania.



Rys. 2. Rozproszony model zaufania

Połączenie równorzędnych podstawowych CA nazywane jest certyfikacją wzajemną (krzyżową) lub siecią PKI. Nie jest to jedyna metoda ustanowienia związku zaufania w sieciach tego typu. Innymi z nich są:

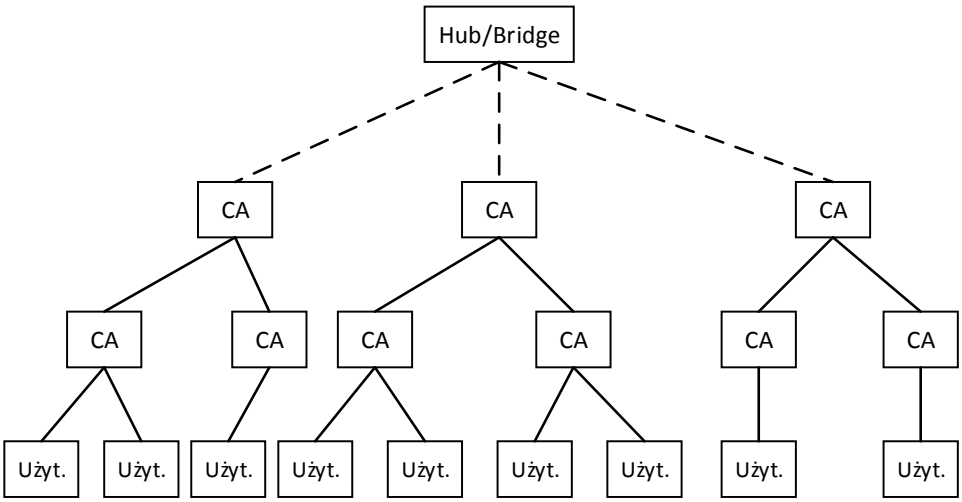
- wzajemne uznanie – uznawanie CA w innych domenach PKI dzięki akredytacji przez uznawaną przez wszystkich trzecią stronę (organ akredytacji);
- lista zaufanych certyfikatów – podpisana lista certyfikatów podstawowego organu certyfikacji, którą administrator obdarza zaufaniem do wybranych celów;
- certyfikat akredytacji – metoda, w której dobrze znane i zaufane CA poręcza za inne CA; wykonywana jest operacja jednokierunkowej certyfikacji wzajemnej, ale bez wprowadzenia hierarchizacji [4].

Jeżeli chodzi o konfiguracje, w jakich spotykana jest rozproszona architektura zaufania, to jedną z podstawowych jest konfiguracja siatki, w której wszystkie podstawowe CA mogą mieć certyfikację wzajemną ze sobą. W szczególności możliwe jest uzyskanie siatki pełnej, w której ustanowiono relacje zaufania w postaci każdy z każdym. Przykładem pełnej siatki jest model ogólny z rysunku 2, na którym każde z głównych CA ma ustanowione związki zaufania z każdym innym CA, co jest oznaczone linią przerywaną. Przypadki pozostałe nazywane są siatką częściową i zawierają tylko te połączenia, które na etapie wdrożenia modelu okazały się potrzebne.

Oprócz siatki, występuje również konfiguracja koncentratora z rozgałęzieniami (ang. *Hub-and-Spoke*, często spotykana jest również nazwa *Bridged CA* – pomostowa), w której każdy podstawowy CA ustanawia certyfikację wzajemną z jednym centralnym CA (pełniącym funkcję koncentratora, często nazywanym również mostem, ang. *bridge*). Centralny CA bywa także nazywany pomostowym, ze względu na funkcję, jaką pełni – łącznika pomiędzy dowolnymi podstawowymi CA. Niektóre źródła definiują ten rodzaj struktury jako oddzielny model zaufania [5]. Ze względu na jego oparcie w architekturze rozproszonej oraz niewielkie modyfikacje w stosunku do jej oryginalnych założeń, zdecydowano się zaklasyfikować go jednak jako podzbiór infrastruktury rozproszonej. Rysunek 3 obrazuje strukturę tego modelu.

Ze struktury przedstawionej na rysunku 3 wyraźnie widoczne jest podobieństwo do architektury rozproszonej, z dokładnością do metody ustanawiania wzajemnego zaufania przez główne CA. W tym miejscu nasuwa się również podobieństwo przedstawionej architektury z modelem hierarchicznym, ze względu na występowanie koncentratora jako węzła znajdującego się najwyżej w hierarchii. Należy zatem podkreślić, że most nie jest korzeniem dla innych węzłów. Różnica jest szczególnie widoczna w przypadku liści drzewa, gdzie w modelu hierarchicznym wszystkie jednostki przechowują kopię klucza głównego CA, traktując je jako swoją kotwicę zaufania, podczas gdy w modelu pomostowym żadna jednostka końcowa nie przechowuje jako takiej kotwicy klucza CA pełniącego funkcję mostu. Liść przechowuje kopię klucza CA z własnej domeny, w razie potrzeby za pomocą

standardowej procedury przetwarzania ścieżki certyfikacyjnej może otrzymać kopię klucza koncentratora po to, aby otrzymać kopię klucza CA pochodzącego z innej domeny, do której należy jednostka, którą liść chce zweryfikować.



Rys. 3. Model z koncentratorem

Jako kolejną modyfikację ww. struktury wprowadza się w literaturze architekturę wielu mostów (ang. *Multiple bridged CA*) [8]. Jej koncepcja opiera się na tym, iż może istnieć wiele koncentratorów, a jedno CA stanowiące korzeń dla swojej domeny może należeć do więcej niż jednego mostu, ale nie ma obowiązku należeć do wszystkich. Model taki pozwala na łączenie ze sobą różnych domen PKI za pomocą różnych ścieżek, zmniejszając problem modelu podstawowego, czyli konieczności zaufania przez wszystkie rozgałęzienia jednemu CA, pełniącemu funkcję koncentratora.

W przypadku bezpośredniej certyfikacji wzajemnej (struktura siatki) znacząco wzrasta liczba połączeń pomiędzy głównymi CA, które poza swoim drzewem (domeną PKI) muszą utrzymywać certyfikaty innych CA. Uznawane jest to za wadę rozwiązania, która stanowi poważną przeszkodę do implementacji rozwiązania, jak również może stać się źródłem sporów, kiedy niektóre z głównych CA znajdują się pod jurysdykcją stron, które z różnych względów (np. politycznych) nie są zainteresowane utrzymywaniem ze sobą relacji, w szczególności nie chcą dokonać certyfikacji wzajemnej [7]. Podobny problem zaobserwować można w przypadku modelu z mostem: również nie wszystkie strony mogą być zainteresowane certyfikacją wzajemną z CA pełniącym funkcję koncentratora. Model pomostowy pozbawiony jest jednak wady związanej z utrzymywaniem wielu połączeń pomiędzy różnymi CA,

w praktyce sprowadzając się do jednego nadmiarowego połączenia z koncentratorem, dla każdego CA będącego korzeniem dla swojej domeny.

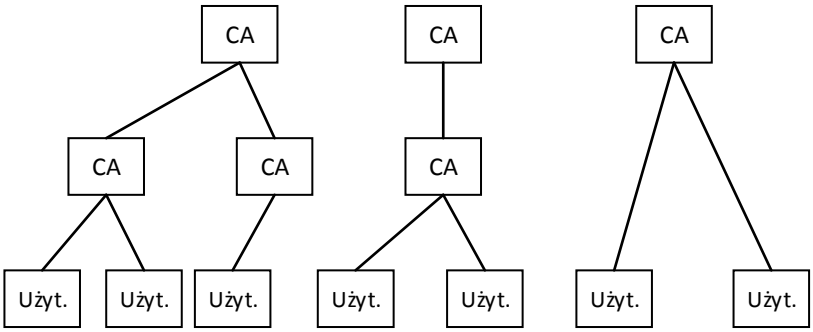
2.5. Model czworokątny

W czworokątnym modelu zaufania występują cztery strony każdej transakcji: subskrybent (użytkownik), CA subskrybenta, strona polegająca na certyfikacie oraz CA tejże strony. Zarówno subskrybent, jak i podmiot polegający na certyfikacie komunikują się tylko ze swoimi CA oraz ze sobą. Każda czynność, która wymaga potwierdzenia prowadzi do tego, że podmiot polegający na certyfikacie korzysta wyłącznie ze swojego CA, które wykonuje całą pracę związaną z kwestiami weryfikacji oraz autoryzacji transakcji w CA subskrybenta. Jest to cecha odróżniająca czworokątny model zaufania od modelu rozproszonego. Ze względu na brak konieczności ujawniania wszystkich informacji pomiędzy podmiotami, między którymi zachodzi transakcja, model ten jest rozwiązaniem chroniącym prywatność jego użytkowników – dane stron udostępniane są tylko swoim CA, które same zatwierdzają i autoryzują transakcję. Z przytoczonego powodu struktura czworokątna znajduje szerokie zastosowanie w różnych formach transakcji elektronicznych [4] – zakupy przez sieć, przelewy bankowe itp.

2.6. Model sieci WWW

Model sieci WWW nazwę zawdzięcza zastosowaniu go w przeglądarkach internetowych. Jego działanie opiera się na preinstalowaniu puli certyfikatów publicznych CA w przeglądarce dostępnej standardowo dla każdego. Pula taka ma za zadanie zdefiniować pierwotny zbiór zaufany danej przeglądarki, działając jako korzeń procesu weryfikacji innych certyfikatów. Niekiedy model ten jest nazywany jako „skonfigurowane i delegowane CA" (ang. *Configured Plus Delegated CAs*) [5]. Pomimo pozornego podobieństwa do architektury rozproszonej, model WWW jest bliski hierarchicznej organizacji modelu zaufania. Zasadniczo każdy producent przeglądarki posiada własny korzeń, którym „certyfikuje" podstawową pulę kluczy publicznych CA, w swoim własnym produkcie. Termin „certyfikuje" celowo został ujęty w cudzysłowie, ponieważ de facto inicjalny zbiór CA jest wbudowany w oprogramowanie, co zapewnia bezpieczne powiązanie pomiędzy nazwą CA oraz jego kluczem publicznym. Widać wtedy, że otrzymywana jest ścisła hierarchia, gdzie producent przeglądarki jest korzeniem całego drzewa, natomiast zbiór podstawowych CA znajduje się na pierwszym poziomie tego drzewa. Przykład takiego modelu widoczny jest na rysunku 4. CA stanowiące korzenie są predefiniowane w przeglądarce i możliwe jest potwierdzenie tożsamości tylko

tych węzłów podrzędnych, które wywodzą się od jednego z certyfikatów CA głównych [4].

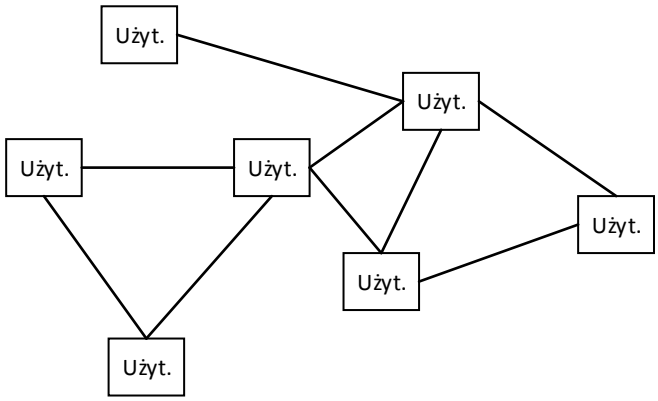


Rys. 4. Model sieci WWW

Niewątpliwą zaletą modelu sieci WWW jest jego prostota użycia oraz wygoda i łatwość współdziałania. Niestety, cechuje się on także wieloma problemami związanymi z bezpieczeństwem. Jednym z nich jest fakt, iż użytkownik przeglądarki automatycznie obdarza zaufaniem wszystkie organy dostarczone wraz z oprogramowaniem. Implikuje to, iż w przypadku gdy jedno z CA podstawowych nie dopełnia obowiązku starannej weryfikacji jednostek które certyfikuje, mimo wszystko użytkownik przeglądarki obdarzy zaufaniem certyfikaty wystawione przez takie CA, nawet jeżeli dojdzie do fałszerstwa – użytkownik nie wie, który z kluczy podstawowych weryfikuje jaki certyfikat pochodny, w związku z czym nie jest w stanie przeciwdziałać takim zachowaniom. Innym problemem jest brak mechanizmu unieważniania kluczy podstawowych. Jeżeli dane CA zostanie przełamane np. poprzez wyciek klucza prywatnego, to model sam w sobie nie daje żadnych możliwości ochrony użytkowników. Obecnie rozwiązywane jest to za pomocą aktualizacji samej przeglądarki, jednak dopóki użytkownik nie przeprowadzi takiej aktualizacji, staje się potencjalnym celem ataku. Aktualizacja takiej przeglądarki niesie ze sobą jednak inne ryzyko – konieczne staje się przeprowadzenie takiej procedury również w sposób bezpieczny, tak aby otrzymać nowe zaufane kopie certyfikatów publicznych CA oraz unieważnić niektóre stare. Jeżeli nie zostaną zachowane odpowiednie środki ostrożności podczas takiego procesu, możliwe staje się dodanie CA, którego instalacja nie była intencją strony wydającej aktualizację. Jednostki końcowe otrzymują w taki sposób certyfikat złośliwego CA, który może być użyty w późniejszym czasie przez atakujących do autoryzacji wykonania złośliwego oprogramowania lub wykonania innych czynności jako pełnoprawny użytkownik (lub oprogramowanie) [5].

2.7. Model oparty na użytkowniku

Model określany jako „użytkownikocentryczny" [4] lub „anarchiczny" [5]. Każdy z użytkowników odpowiedzialny jest za decyzję o tym, czy certyfikat przedstawiony przez drugą stronę może uznać za zaufany, czy też należy go odrzucić. Decyzja podejmowana przez użytkownika może zależeć od wielu czynników (np. możliwe jest wspomaganie mechanizmem ratingowym, opinią innych użytkowników), może być również w pełni subiektywna i zasadniczo nie określają tego żadne normy.



Rys. 5. Model oparty na użytkowniku

Zaufanie oparte na użytkowniku sprawdza się w przypadku społeczności o niezdefiniowanej strukturze, pod warunkiem odpowiedniej świadomości technicznej osób w niej uczestniczących. Model ten został wdrożony z powodzeniem na świecie, a jako największy działający przykład należy podać PGP (*Pretty Good Privacy*), wykorzystywane przez miliony osób w Internecie [7]. Wśród wielu zalet posiada on jednak sporo wad, przede wszystkim brak kontroli nad tym, kogo poszczególny podmiot obdarza zaufaniem oraz konieczność rozwinięcia technicznego użytkowników takiego systemu. W związku z tym, model oparty o użytkownika nie nadaje się do zastosowania w środowiskach korporacyjnych, finansowych, administracyjnych i wielu innych, gdzie wymagana jest skrupulatna kontrola zaufania.

2.8. Modele hybrydowe

Wymienione powyżej modele zaufania stanowią pewną podstawę, na której w dużej mierze wykształciły się modele hybrydowe, będące połączeniem jednego lub więcej omawianych modeli. Część z nich istnieje tylko teoretycznie, na potrzeby analizy ich działania i koncepcji, z kolei wiele doczekało się

pojedynczych implementacji. Brak spójności w dziedzinie PKI oraz koncepcji zaufania wynika z faktu, iż ciężko jest wybrać jeden model i zastosować go wszędzie, gdzie to tylko możliwe. Naturalne jest to, że do różnych potrzeb wybierane są różne podejścia, a zatem i różne koncepcje realizacji zarządzania tożsamością cyfrową.

Bardzo wiele modeli hybrydowych opiera się na niewielkich modyfikacjach obecnie istniejących. W szczególności często dodawane są nowe relacje, jakie mogą zachodzić pomiędzy konkretnymi węzłami (np. pomiędzy CA). Przykładami mogą być modele „Up-Cross-Down" [5], gdzie relacja certyfikacji jest relacją nie ograniczoną tylko do certyfikacji podmiotów podrzędnych (nie tylko węzły nadrzędne certyfikują węzły podrzędne, ale możliwa jest sytuacja odwrotna), czy też „Flexible Bottom-Up" [5], które wykorzystuje rozszerzenia specyficzne dla PKIX (PKI dla X.509).

W praktyce organizacje, które wdrażają u siebie infrastrukturę klucza publicznego, często korzystają z modeli hybrydowych, biorąc to, co jest im potrzebne z istniejących modeli i tworząc swoje rozwiązanie, dopasowane do konkretnych warunków.

3. Podsumowanie

Zaufanie wewnątrz PKI propagowane jest za pomocą konkretnych struktur modeli zaufania. Do najważniejszych, z punktu widzenia tworzenia kolejnych modeli, należą model hierarchiczny oraz rozproszony. W każdym innym modelu możliwe jest dostrzeżenie podobieństw do jednego z dwóch powyższych, zarówno w strukturze, jak i w schemacie działania. Na tym tle wyróżnia się model zaufania oparty na użytkowniku, który nie pasuje do żadnego z dwóch głównych modeli. Bliżej mu do architektury rozproszonej, jednak brak jakiejkolwiek struktury oraz brak możliwości tworzenia przez węzły własnych podstruktur nie pozwala go sklasyfikować jako bezpośrednio wywodzący się z rozproszonej siatki.

Przy aktualnym stanie wiedzy oraz dostępnych modelach nie jest możliwe określenie jednego, pasującego do każdej sytuacji modelu. Powodem wytworzenia tak dużej liczby struktur jest niemożność stworzenia uniwersalnego modelu, który byłby dostatecznie elastyczny oraz skalowalny, aby dobrze pasować do małych oraz dużych PKI, sprawdzać się w pojedynczych firmach, jak również w korporacjach, czy spełniać swoją funkcję nawet na poziomie całego kraju. Funkcjonowanie wielu różnych domen PKI, często nawet w obrębie jednej dużej korporacji, powoduje problemy natury interoperacyjności użytkowników pomiędzy takimi domenami. Częściowo problem jest rozwiązywany przez wprowadzenie mostów oraz modelu brigde CA, jednakże

rozwiązanie takie może funkcjonować poprawnie tylko w określonych warunkach, gdzie podstawową kwestią jest ustalenie zaufania do mostu. W przypadkach gdy domeny PKI, które chce się połączyć, znajdują się pod jurysdykcją różnych korporacji lub na przykład różnych krajów, problem chęci certyfikacji wzajemnej z jednym CA będącym mostem, jak również ustalenie położenia oraz odpowiedzialnego za kontrolę takiego CA staje się wyzwaniem i często uniemożliwia połączenie takich domen.

Każdy model niesie ze sobą pewną ilość zalet oraz wad, które często wymieniają się pomiędzy modelami. Głównym problemem wielu przedstawionych struktur (hierarchia, WWW) jest dystrybucja kluczy głównych CA, która wymaga ustanowienia nowych kanałów bezpiecznej łączności (dla inicjalnego przekazania kluczy) bądź preinstalacji określonego zestawu certyfikatów przed przekazaniem urządzenia (lub oprogramowania – model WWW) do jednostki końcowej. Wprowadza to oczywisty problem związany z unieważnianiem oraz wymianą takiego inicjalnego zestawu certyfikatów dla wszystkich jednostek, stanowiąc lukę w bezpieczeństwie modelu. Podobna luka istnieje w przypadku modelu hierarchicznego, gdzie przełamanie zaledwie jednego klucza (root CA) powoduje brak zaufania w całym modelu. Na tle bezpieczeństwa lepiej prezentują się modele rozproszone, gdzie przełamanie nawet jednego z głównych CA powoduje utratę zaufania tylko do jednej gałęzi struktury, a nie do całego modelu. Mają one jednak wady opisane w poprzednim akapicie, których nie doświadcza się w modelach hierarchicznych.

Prace badawcze nad rozwojem kolejnych modeli zaufania oraz usprawnienia istniejących, przez wprowadzanie do nich rozszerzeń (np. w postaci kolejnych relacji, protokołów komunikacyjnych), jest w pełni zasadne, biorąc pod uwagę niedoskonałości rozwiązań istniejących obecnie. Poszukiwanie rozwiązań podnoszących poziom zaufania pomiędzy uczestnikami transakcji oraz podnoszących ogólne bezpieczeństwo wewnątrz struktur PKI ma duże znaczenie dla funkcjonowania wielu firm, w szczególności organizacji wirtualnych, których działanie opiera się na komunikacji poprzez sieć swoich zdecentralizowanych komórek. Również instytucje, w szczególności te obdarzane zaufaniem publicznym, muszą wciąż rozwijać swoje systemy, zapewniając najwyższy poziom bezpieczeństwa przetwarzanych i przesyłanych danych.

Literatura

[1] *PN-I-02000. Technika informatyczna. Zabezpieczenia w systemach informatycznych. Terminologia*, Polski Komitet Normalizacyjny, 2002.

[2] *ITU-T Recommendation X.1252* (04/2010).

[3] GERCK E., *Trust as Qualified Reliance on Information*, [w:] *The COOK Report on Internet Protocol*, 2002, pp. 20-24.

[4] ADAMS C., LLOYD S., *PKI podstawy i zasady działania. Koncepcje, standardy i wdrażanie infrastruktury kluczy publicznych*, Wyd. Naukowe PWN, Warszawa, 2007.

[5] PERLMAN R., *An Overview of PKI Trust Model*, [w:] *IEEE Network*, November/December, 1999, pp. 38-43.

[6] ITU-T Recommendation ITU-T X.509 (10/2012).

[7] JØSANG A., *PKI Trust Models*, [w:] *Theory and Practice of Cryptography Solutions for Secure Information Systems* (CRYPSIS), IGI Global, May, 2013.

[8] MOSES T., *PKI trust models*, IT University of Copenhagen, 2003 (http://www.itu.dk/courses/DSK/E2003/DOCS/PKI_Trust_models.pdf).

PKI trust models overview

ABSTRACT: The paper considers different Public Key Infrastructure (PKI) trust models. The PKI can be described as set of technologies, policies, procedures and people, which allows to manage public key certificates as well as managing trust between two or more parties. In order to manage trust relations so-called 'trust models' structures have been introduced. Those models can be divided into several groups, depending on their common features, particularly on similar structure and a way of handling relations between their nodes. This overview is aimed on presenting currently used trust models including their key features and short discussion on their advantages and disadvantages.

KEYWORDS: PKI, public key infrastructure, trust, trust models, X.509

Praca wpłynęła do redakcji: 20.12.2014 r.

Wielokierunkowy test wskazywania – norma ISO 9241-9 – przegląd badań

Antoni M. DONIGIEWICZ

Instytut Teleinformatyki i Automatyki WAT
ul. Gen. S. Kaliskiego 2, 00-908 Warszawa
a.donigiewicz@ita.wat.edu.pl

STRESZCZENIE: W artykule przedstawiono wielokierunkowy test wskazywania stosowany do oceny jakości wprowadzania informacji za pomocą urządzeń wskazujących. Opis testu oparto na normie ISO 9241-9. Test może być podstawą oceny i oszacowania jakości wprowadzania informacji przez użytkownika. Przedstawiono wybrane wyniki badań dostępne w literaturze.

SŁOWA KLUCZOWE: testy urządzeń wskazujących, wielokierunkowy test wskazywania, prawo Fittsa, norma ISO 9241-9

1. Wprowadzenie

W artykule [4] przedstawiono opis jednokierunkowego testu wskazywania oraz wybrane badania opisywane w literaturze wykorzystujące ten test do oceny jakości działania użytkownika. W niniejszym artykule przedstawiono wielokierunkowy test wskazywania opisany w normie [6] oraz wybrane badania opisywane w literaturze wykorzystujące ten test do oceny realizacji zadań za pomocą urządzeń wskazujących. Część przedstawionych badań obejmuje również realizację zadań z wykorzystaniem gestów, w których stosowano częściowo zmodyfikowany wielokierunkowy test wskazywania.

W artykule opis badań i wyniki obejmują zasadniczo czas od obowiązywania normy ISO 9241-9. Dla pełnego spojrzenia na przedstawioną problematykę przedstawiono skrótowo również badania dla okresu przed oficjalnym wprowadzeniem normy [6]. W czasie obowiązywania normy [6] wyróżniono dwa okresy: okres stosowania klasycznych warunków i typowych urządzeń (można uważać, że trwał on do ok. 2006 roku) i okres stosowania zmiennych warunków i nietypowych urządzeń (od roku 2007). Oczywiście

granicy pomiędzy tymi okresami nie należy traktować sztywno. Należy jednak zwrócić uwagę na różnice pomiędzy tymi okresami pod względem warunków stosowania testu oraz właśnie nietypowych urządzeń wprowadzania. Na końcu każdego z wyróżnionych okresów wskazano na badania częściowo związane z wielokierunkowym testem wskazywania, ale bez szczegółowego przedstawienia wyników badań.

2. Wielokierunkowy test wskazywania

2.1. Procedura testowania

Testy dotyczące wskazujących urządzeń wprowadzania, opisywane w literaturze [6], [20], dotyczą oceny realizacji następujących zadań podstawowych:

- wskazywanie;
- wybieranie;
- ciągnięcie (przeciąganie);
- śledzenie;
- wprowadzanie swobodne.

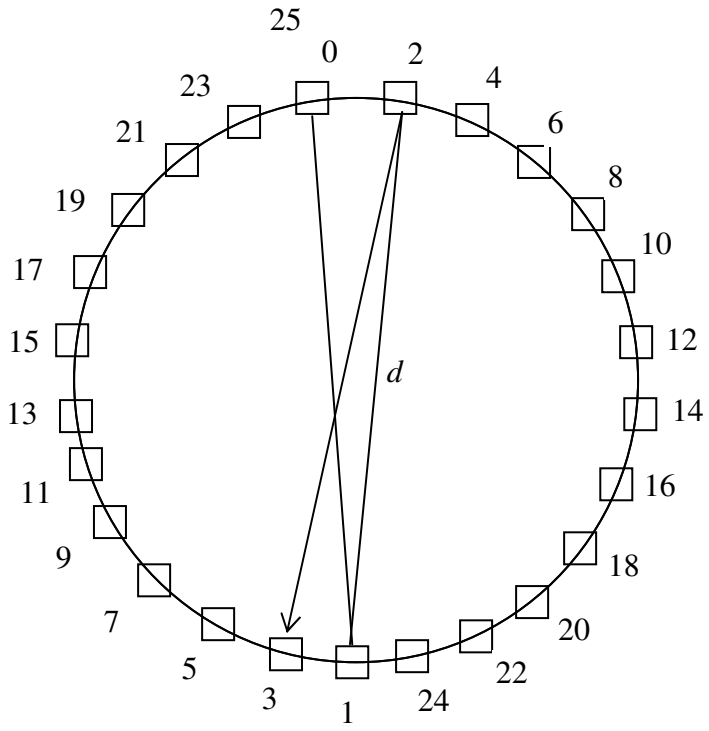
Wielokierunkowy test wskazywania może być wykorzystany do oceny jakości wskazywania w różnych kierunkach. Przykładowo może być stosowany do oceny jakości wykonywania następujących zadań:

a) umieszczanie wskaźnika w różnych obszarach na ekranie;

b) wybór komórek na arkuszu kalkulacyjnym;

c) wybór (wskazywanie) losowo umieszczonych ikon.

W ramach procedury testowania należy przemieszczać wskaźnik w poprzek koła do kolejno ponumerowanych obiektów (kwadratów) (rys. 1). Kwadraty powinny być równomiernie rozmieszczone na brzegu okręgu, tak aby przemieszczenie wskaźnika było w przybliżeniu równe średnicy okręgu. Kwadrat, do którego wskaźnik powinien być przesunięty, powinien być podświetlony. Każda seria testowa rozpoczyna się wówczas, gdy zostanie wskazany kwadrat najwyżej położony i kończy się, kiedy sekwencja wskazań zostanie zakończona (wskazanie najwyżej położonego kwadratu). Test ten powinien być związany ze zmianą zakresu trudności poprzez zmianę średnicy okręgu pomiędzy próbami (kwadraty na okręgu pozostają bez zmian). Obiektami wskazywanymi na okręgu w teście wzorcowym (norma [6]) są kwadraty, natomiast w prowadzonych badaniach, jako obiekty stosowano czasami kółka.



Rys. 1. Ilustracja wielokierunkowego testu wskazywania (na podstawie [6])

2.2. Wyznaczanie charakterystyk urządzenia wejściowego

Po przeprowadzeniu testu wyznaczane są następujące charakterystyki [6], [13], [20].

Skuteczna szerokość obiektu (w_e) jest to szerokość rozproszenia wybranych współrzędnych uzyskana w wyniku wskazywania (klikania) podczas testu. Wielkość ta obliczana jest z zależności [6], [13]:

$$w_e = 4{,}133\, s_x , \tag{1}$$

gdzie: s_x – odchylenie standardowe współrzędnych w kierunku kontynuowania ruchu (np. zgodnie z osią x).

Wskaźnik trudności *ID* jest miarą precyzji użytkownika wymaganej w zadaniu. Wyrażany jest w bitach. Dla zadań wskazywania, wyboru lub przeciągania wyznaczany jest z zależności [6], [19]:

$$ID = \log_2 \frac{d + w}{w}, \tag{2}$$

gdzie: w – wielkość obiektu (w teście wielokierunkowym bok kwadratu – rys. 1),

d – odległość ruchu urządzenia (w teście wielokierunkowym średnica okręgu jak na rys. 1).

Skuteczny wskaźnik trudności ID_e dla zadań wskazywania, wyboru lub przeciągania wyznaczany jest z zależności [6], [13]:

$$ID_e = \log_2 \frac{d + w_e}{w_e}, \tag{3}$$

gdzie: w_e, d – jak w zależności (1) i (2) odpowiednio.

Przepustowość wejściowa P_w dla zadań wskazywania, wyboru, przeciągania i śledzenia wyznaczana jest z zależności [6], [13]:

$$P_w = \frac{ID_e}{t_m}, \tag{4}$$

gdzie: ID_e – skuteczny wskaźnik trudności dla zadania;

t_m – czas przemieszczenia (ruchu) wyznaczany od rozpoczęcia ruchu urządzenia wejściowego do wskazania (lub wybrania) obiektu.

Szybkość przemieszczania wskaźnika V_p jest to średnia szybkość, z jaką użytkownik wykonuje wskazania obiektów i wyznaczana jest z zależności [6], [13]:

$$V_p = \frac{d}{t_m} \tag{5}$$

gdzie: d – jak na rys. 1;

t_m – jak w zależności (4).

2.3. Wielokierunkowy test wskazywania stosowany w testach urządzeń

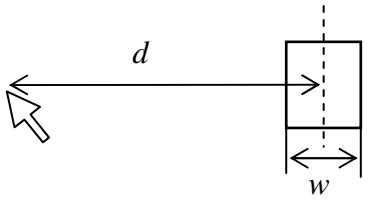
Omawiany wielokierunkowy test wskazywania stosowany był jako test oceny jakości realizacji zadań za pomocą różnych urządzeń wejścia-wyjścia. Test był stosowany również przed publikacją normy ISO 9241-9 [5], [22]. Od chwili publikacji normy (rok 2000) test z różnymi zmianami stosowany był w badaniach wielokrotnie.

Głównym celem prowadzonych badań było i jest wyznaczenie podstawowej charakterystyki, jaką jest czas przemieszczenia (ruchu) wskaźnika urządzenia wejściowego na ekranie. Badania najczęściej umożliwiają wyznaczenie parametrów równania opisującego czas ruchu wskaźnika do celu na ekranie, poruszanego przez użytkownika za pomocą urządzenia wejściowego. Równanie to nazywane prawem Fittsa (rys. 2) przedstawione jest poniżej w postaci najczęściej spotykanej [4], [7], [19], [20], [26]:

$$t_r = a + b\,ID\,, \tag{6}$$

gdzie: a, b – stałe wyznaczane doświadczalnie;

ID – wskaźnik trudności (najczęściej w postaci jak w zależności (2)).



Rys. 2. Ilustracja prawa Fittsa [1], [14], [19], [20], [27]

Zamiast czasu przemieszczenia wskaźnika urządzenia wejściowego wyznaczana jest czasami przepustowość wejściowa (jak w zależności (4)). Drugim, najczęściej wyznaczanym parametrem jest procent błędów (stopa błędów) w trafieniu w obiekt w czasie testu.

W badaniach, w których wykorzystywano wielokierunkowy test wskazywania, można wyróżnić trzy okresy:

- okres przed oficjalnym wprowadzeniem normy,
- okres stosowania klasycznych warunków i typowych urządzeń,
- okres stosowania zmiennych warunków i nietypowych urządzeń.

Poniżej przedstawiono wybrane wyniki badań, które były opublikowane w wymienionych okresach czasu. Czas obowiązywania normy [6] podzielono na dwa okresy. Pierwszy z nich – okres stosowania klasycznych warunków i typowych urządzeń – trwał do końca 2006 roku włącznie.

3. Wyniki badań

3.1. Okres przed oficjalnym wprowadzeniem normy ISO 9241-9

Okres przed wprowadzeniem normy ISO 9241-9 to okres do roku 2000 (część publikacji wskazuje na rok 2002 jako rok oficjalnego wprowadzenia normy ISO 9241-9, np. [20]). W tym okresie stosowano już zasady (test) opisane w normie. Typowymi badaniami stosowanymi w tym okresie były badania przedstawione w pracy [5].

Badania były szersze niż badania związane z wielokierunkowym testem wskazywania. Obejmowały również wskazywanie jednokierunkowe, przeciąganie, wprowadzanie z wolnej ręki (odręcznie pisane znaki) i uchwycenie urządzenia oraz parkowanie [5].

Stosowane urządzenia

Urządzenia wykorzystywane w tych badaniach to tzw. minijoystick (Trackpointe III na klawiaturze pomiędzy klawiszami „G" i „H") i touchpad (Touchpad 2 model 400 podłączony przez port PS/2). Wykorzystano laptop IBM Thinkpade wyposażony w oddzielny monitor z kolorowym wyświetlaczem 21-calowym [5].

Badani użytkownicy

Badaniami objęto 24 osoby, po dwanaście dla każdego urządzenia. Dla touchpada wszyscy użytkownicy byli praworęczni. Dla joysticka jedenastu użytkowników praworęcznych i jeden leworęczny. Wszyscy użytkownicy mieli wcześniejsze doświadczenie w pracy na komputerze i bogate doświadczenie w pracy za pomocą myszy. Użytkownicy byli przydzieleni do urządzenia, z którym nie mieli wcześniejszego doświadczenia w użytkowaniu.

Procedura badawcza

Dla wielokierunkowego testu wskazywania zastosowano następujące wielkości związane z wyświetlanymi obiektami [5]:

- szerokość obiektu (2 mm, 5 mm, 10 mm),
- odległość obiektu (40 mm, 80 mm, 160 mm),
- kąt położenia obiektu (0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°).

Badanie rozpoczynało się, gdy użytkownik kliknął w kwadrat początkowy (w centrum okna aplikacji), a kończyło się, gdy użytkownik kliknął w kole celu wyświetlonym w określonym miejscu. Czas pomiędzy tymi kliknięciami był rejestrowany jako czas próby. Wskaźnik był automatycznie przenoszony na środek kwadratu początkowego na końcu każdej próby. Kombinacje szerokości, odległości i kąta były stosowane losowo.

Wyniki badań i komentarze

Każda osoba wykonała 72 badania w 10 seriach (łącznie 720 prób) [5]. Średni czas ruchu dla joysticka to 1,975 s, z odchyleniem standardowym wynoszącym 0,601 s. Dla touchpada średni czas ruchu był równy 2,382 s, a odchylenie standardowe wynosiło 0,802 s. Różnice w średnim czasie ruchu były istotne statystycznie. Z tego wywnioskowano, że czas wskazywania dla joysticka jest o 17% mniejszy od średniej. Stopa błędów była równa 2,1% dla joysticka i 5,4% dla touchpada. Różnice w stopie błędów były istotne statystycznie. Przepustowość wyznaczona dla joysticka była równa 2,15 bitu/s (odch. stand. = 0,40), natomiast dla touchpada 1,70 bitu/s (odch. stand. = 0,53). Wyniki badań porównano z wynikami dla myszy komputerowej.

Po eksperymencie uczestnicy oceniali za pomocą ankiety urządzenia pod względem wygody użycia, działania, zmęczenia i użyteczności [5]. Ankieta wykazała brak różnic w odpowiedziach i istotną różnicę statystyczną tylko pytań dotyczących siły potrzebnej do użycia joysticka, który wymagał nieco większej siły. W badaniach nie wyznaczono wielkości stałych występujących w równaniu prawa Fittsa (por. wzór (6)).

W artykule zwraca się uwagę, że norma [6] zaleca co najmniej 25 użytkowników przy badaniach związanych z testami jakości działania użytkownika. Należy zwrócić uwagę, że stosowany w badaniach test nie był w pełni zgodny z normą [6].

Wcześniejszymi badaniami, które w pewnym zakresie były zgodne z procedurą opisaną w normie [6] są badania przedstawione w pracy [22]. Badania wykonane zostały dla ośmiu kierunków położenia obiektów, ale nie są w pełni zgodne z warunkami przedstawionymi w normie.

3.2. Okres stosowania klasycznych warunków i typowych urządzeń

Badania opisywane poniżej są właściwie na granicy dwóch okresów: okresu przed wprowadzeniem normy i okresu stosowania klasycznych warunków badań. W badaniach [15], których wyniki przedstawiono w 2001 roku, zaproponowano nowe miary zmienności ruchu (odchylenia ruchu od linii prostej w kierunku obiektu). Propozycje te były następujące:

- zmienność ruchu (odchylenie standardowe współrzędnej *y* wybranych punktów od średniej): $MV = \sqrt{\frac{\sum(y_i-\bar{y})^2}{n-1}}$,
- błąd ruchu (średnie odchylenie wybranych punktów od współrzędnej zadania – celu): $ME = \frac{\sum|y_i|}{n}$,
- przesunięcie ruchu: $MO = \bar{y}$.

Wyróżniono również typowe zmiany w ścieżce wskaźnika urządzenia w ruchu do celu:

- dwukrotne wskazanie celu (TRE),
- przekroczenie współrzędnej *y* zadania (celu) (TAC),
- zmiana kierunku ruchu (MDC),
- ortogonalna zmiana kierunku ruchu (ODC).

Stosowane urządzenia

Eksperyment przeprowadzono na komputerze klasy Pentium z systemem Windows 98. Wyjściowym urządzeniem był 17-calowy monitor. Wejściowymi urządzeniami były cztery następujące urządzenia wskazujące [15]:

– mysz (Logitech *FirstMouse+*),

– trackball (Logitech *TrackMan Marble*),

– joystick (Interlink *DeskStick*),

– touchpad (Touché *Touchpad*).

Badani użytkownicy

Użytkownikami było 12 osób. Byli oni losowo przydzieleni do jednej z czterech grup (3 użytkowników na grupę). Każdy z użytkowników był badany z użyciem wszystkich urządzeń. Kolejność użycia urządzeń różniła się dla każdej grupy. Przed przystąpieniem do badania uczestnicy zostali poinformowani o celu eksperymentu. Zadanie było klasyczne – wielokierunkowy test wskazywania zgodny z ISO 9241-9 [6]. Zadanie zademonstrowano użytkownikom i przed badaniem wykonano sekwencję rozgrzewki [15].

Procedura badawcza

W badaniach użyto 16 kółek – celi ułożonych na kole (liczba celi inna niż w normie [6]). Średnica koła była równa 400 pikseli (180 mm), natomiast średnica każdego celu 30 pikseli (13 mm). Użyty został tylko jeden poziom trudności zadania o wskaźniku trudności równym 3,8 bitu. Sekwencja prób była zgodna z normą [6]. Każdy „następny" cel był wskazywany fioletowym krzyżykiem, który przenoszony był od celu do celu, zgodnie z kolejnością badania. Sygnalizowano dźwiękiem każdy wybór ze wskaźnikiem poza celem. Badania obejmowały 10 bloków. Blok zawierał 5 sekwencji badania [15].

Wyniki badań i komentarze

Jeśli chodzi o czas ruchu, to mysz była najszybszym urządzeniem, natomiast joystick najwolniejszym. Przepustowość wynosiła 4,9 bitu/s dla myszy, 3 bity/s dla trackballa, 1,8 bitu/s dla joysticka, natomiast 2,9 bitu/s dla touchpada. Różnice w wartościach były istotne statystycznie z wyjątkiem trackballa oraz touchpada.

Stopa błędów była równa 9,4% dla myszy, 8,6% dla trackballa, 9,0% dla joysticka, a 7,0% dla touchpada. Różnice w wartościach nie były istotne statystycznie.

Tabela 1 przedstawia wartości średnie (Śr.) i odchylenia standardowe (Odch.) siedmiu miar dokładności dla czterech badanych urządzeń. Oczywiście w odniesieniu do wszystkich miar, niższe wartości są lepsze. Jednostki w tab. 1 są „średnimi obliczonymi na próbę" dla TRE, TAC, MDC i ODC oraz „pikselami" dla MV, ME i MO, gdzie 1 piksel = 0,43 mm mierzone na ekranie.

Tab. 1. Wartości średnie (Śr.) i odchylenia standardowe (Odch.) miar dokładności dla czterech badanych urządzeń [15]

Zmienna	Mysz		Trackball		Joystick		Touchpad	
	Śr.	Odch.	Śr.	Odch.	Śr.	Odch.	Śr.	Odch.
TRE	0,07	0,04	0,26	0,13	0,33	0,08	0,15	0,04
TAC	1,7	0,2	2,2	0,4	2,0	0,3	1,64	0,19
MDC	3,6	1,0	5,7	1,6	6,1	3,8	3,6	0,7
ODC	0,8	0,4	1,8	0,6	1,5	0,9	0,8	0,2
MV	10,5	3,9	15,9	2,5	17,6	3,8	11,7	2,4
ME	11,6	4,7	16,5	3,6	18,7	3,5	13,2	2,5
MO	2,5	1,0	3,4	0,8	5,1	1,8	3,9	2,4

Różnice w przepustowości urządzeń w porównaniu do innych badań wynikają z różnych testowanych produktów. W badaniach nie wyznaczono wartości stałych występujących w równaniu prawa Fittsa (por. wzór (6)). Należy zwrócić uwagę, że stosowana procedura badawcza nie w pełni odpowiada procedurze opisanej w normie [6]. Występuje różnica w liczbie obiektów (celi) położonych na okręgu w opisywanych badaniach w stosunku do normy [6].

W początkowym okresie obowiązywania normy prowadzono badania w zasadzie wg klasycznych zasad, choć stosowano pewne modyfikacje. Modyfikacje te dotyczyły przykładowo liczby obiektów (celów) występujących w teście (na okręgu koła) lub wielkości tych obiektów [9].

Wskazane badania [9] dotyczyły wykorzystania techniki i urządzenia TouchGrid (urządzenie typu touchpad), które umożliwia wybranie grupy obiektów, a później szczegółów (tzw. wskaźniki komórkowe). Wybranie obiektu docelowego wiąże się z trzema puknięciami (tap) z towarzyszącymi im zmianami – zagęszczanie kraty na fragmencie figury. Ostatni wybór celu składa się z trzeciego puknięcia (tap) w określony pasek (kratkę). Regiony touchpada są rekurencyjnie mapowane do mniejszych regionów zobrazowania i tym samym umożliwiają precyzyjne wskazanie.

Podstawą badań była sugestia, że czas wyboru obiektu za pomocą TouchGrid jest liniową funkcją wymaganej liczby puknięć (tap).

Stosowane urządzenia

W eksperymencie [9] użyto TouchGrid i typowego touchpada. Aplikację uruchamiano na laptopie 1 GHz z touchpadem $6 \times 4{,}5$ cm i z 15-calowym ekranem.

Badani użytkownicy

Sześć badanych osób (4 kobiety i 2 mężczyzn) w wieku od 21 do 56 lat (średnia 31,7 lat) używało TouchGrid i typowego touchpada. Wszyscy używali touchpada wcześniej. Pięć osób było praworęcznych i jedna leworęczna.

Procedura badawcza

Podstawą do badań był wielokierunkowy test wskazywania [6]. Badania prowadzono, zmieniając następujące parametry:

– liczba celów (3 poziomy): 8, 24 albo 48 celów równo rozłożonych na obwodzie koła (liczba celi inna niż w normie [6]);
– odległość do celu (2 poziomy): cele były ułożone albo na kole o średnicy 12 cm, albo na kole o średnicy 22 cm;
– szerokość celu (2 poziomy): małe cele miały średnicę 2 mm, duże cele średnicę 6 mm.

Badania każdej osoby obejmowały trzy bloki po 12 zadań. Każde zadanie obejmowało po 20 prób. Połowa osób badanych w pierwszej kolejności wykorzystywała TouchGrid, a druga połowa typowy touchpad. Użytkownicy używali tylko jednej ręki podczas badań. W badaniach następny cel pojawiał się dopiero, gdy poprzedni cel został poprawnie wybrany.

Wyniki badań i komentarze

Wyznaczono czas wyboru obiektu za pomocą TouchGrid jako liniową funkcję wymaganej liczby puknięć do wybrania celu w postaci [9]:

$$T_{TG} = 212 + 842\ N \text{ [ms]},$$

gdzie: N – wymagana liczba puknięć do wybrania celu.

Wyznaczono wielkości stałe występujące w równaniu prawa Fittsa (por. wzór (6)) dla touchpada:

$$t_r = -657 + 469\ ID \text{ [ms]},$$

gdzie: ID – wskaźnik trudności (jak w zależności (6)).

Dla wszystkich zmienianych parametrów (szerokość celu, odległość do celu i liczba celów) badania wykazały, że czas wyboru celu za pomocą touchpada jest większy niż za pomocą TouchGrid. Jedynie dla 48 celów czas wyboru celu za pomocą TouchGrid był większy niż za pomocą touchpada.

Biorąc pod uwagę stopę błędów (średni błąd standardowy), dla wszystkich zmienianych parametrów (szerokość celu, odległość do celu i liczba celów) badania wykazały stopę błędów dla TouchGrid wyższą niż dla touchpada (w dwóch przypadkach stopa błędów dla obu urządzeń była równa).

Modyfikacje warunków badania, z jakimi można się spotkać w publikacjach, dotyczyły – poza liczbą obiektów (celów) występujących w teście (na okręgu koła) lub wielkości tych obiektów – również zmian sposobu pozyskania (wprowadzenia) obiektów [3].

Badania, które przytoczymy [3], dotyczyły czasu pozyskania obiektu w interfejsie typu „rybie oko" (ang. *Fisheye*) – powiększające się obiekty przy zbliżającym się wskaźniku jak w interfejsie MacOS X. W pracy [3] przedstawiono wyniki dwóch eksperymentów. Pierwszy dotyczył porównania skuteczności różnych form podświetlania obiektu. W drugim eksperymencie badano wydajność pozyskania obiektów (celów), które rozszerzają się wizualnie lub następuje rozszerzenie wizualne wraz z pewną przestrzenią otaczającą (rozszerzenie przestrzeni ruchu).

Stosowane urządzenia – eksperyment I

Eksperyment pierwszy był przeprowadzony na komputerze Intel Pentium 4, 2,8 GHz z systemem Linux Fedora Core 3. Grafikę zapewniała karta graficzna GeForce FX5200. Wykorzystano 19-calowy monitor Compaq z rozdzielczością 1600×1200 (częstotliwość odświeżania monitora 75 Hz). Urządzeniem wejściowym była mysz Labtec z trzema przyciskami.

Badani użytkownicy – eksperyment I

Badanych było 16 użytkowników, wszyscy praworęczni (15 mężczyzn i jedna kobieta) [3].

Procedura badawcza – eksperyment I

Zadania uczestników związane były z wyborem kołowych obiektów (celi) zgodnym z testem wielokierunkowym [6]. Średnica koła obiektów była równa 512 pikseli, natomiast średnica celu zmieniana – 6, 10, 24 i 64 pikseli. Jeśli włączone było rozszerzanie wizualne, średnica celów podwajała się, gdy wskaźnik znajdował się nad obiektem (celem). Następny cel był wskazywany przez niebieskie zabarwienie, wszystkie inne obiekty były szare. Udane wskazanie obiektu sygnalizowano przez zmianę koloru obiektu na zielony, natomiast nieudane wskazanie obiektu powodowało zmianę koloru elementu na czerwony. Po każdym udanym lub nieudanym wskazaniu po upływie 500 ms kolejny cel był podświetlany. Gdy rozszerzanie wizualne było wyłączone, cele pozostawały statyczne (niezmienne). Stosowano kombinacje – wizualne rozszerzanie i podświetlanie, co uniemożliwiało uczestnikom przewidywanie zachowania celu [3].

Wyniki badań i komentarze – eksperyment I

Przeprowadzono 4096 prób (16 użytkowników, 16 bloków badań i 16 wyborów). Zbadany ogólny poziom błędów był równy 4,3%. Wyniki szczegółowe [3]: rozszerzanie wizualne wyłączone i bez podświetlania obiektu poziom błędów 4,2%, rozszerzanie wizualne wyłączone i z podświetlaniem obiektu 3,5%, rozszerzanie wizualne włączone i bez podświetlania obiektu 5,1%, rozszerzanie wizualne włączone z podświetlaniem obiektu 4,3%.

Średni czas wprowadzenia (pozyskania) obiektu (celu) w badaniach dla prób zakończonych poprawnie to 0,93 s (odch. stand. 0,33 s). Jak wynika z badań, rozszerzenie wizualne miało istotny wpływ na średni czas wprowadzenia obiektu. Przy rozszerzeniu wizualnym był on równy 0,917 s (odch. stand. 0,31 s) i 0,945 s (odch. stand. 0,35 s) bez rozszerzenia wizualnego. Podświetlanie obiektu nie miało istotnego wpływu na średni czas wprowadzenia obiektu – 0,928 s (odch. stand. 0,33 s) przy podświetlaniu i 0,934 s (odch. stand. 0,33 s) dla braku podświetlania. Wielkość średnicy celu miała istotny wpływ na średni czas wprowadzenia obiektu (od 0,58 s przy średnicy 64 pikseli do 1,33 s przy 6-pikselowych celach).

Stosowane urządzenia – eksperyment II

Urządzenia były identyczne jak w poprzednim eksperymencie, z tym wyjątkiem, że mysz zastąpiono bezprzewodową myszą mechaniczną Logitech. Aplikację uruchamiano w oknie 700×700 pikseli.

Badani użytkownicy – eksperyment II

Jak w eksperymencie I.

Procedura badawcza – eksperyment II

W drugim eksperymencie badano wydajność pozyskania obiektów (celów), które rozszerzają się wizualnie lub następuje rozszerzenie wizualne wraz z pewną przestrzenią otaczającą (rozszerzenie przestrzeni ruchu) [3]. Rozszerzenie przestrzeni ruchu obejmowało po jednym obiekcie otaczającym w każdą stronę. W eksperymencie zastosowano zmienny typ rozszerzania obiektu: statyczny (w celach porównawczych), rozszerzanie wizualne obiektów i rozszerzanie przestrzeni ruchu. Średnica koła obiektów była równa 512 pikseli jak w eksperymencie I. Zmieniano również szerokość obiektów – 6, 12, 24 i 48 pikseli. Jeśli włączone było rozszerzanie wizualne, średnica celów podwajała się, gdy wskaźnik znajdował się nad obiektem (celem). Wszystkie obiekty na kole celów były czerwone, natomiast następny cel do wyboru był sygnalizowany poprzez czarną obwódkę. Poprawne wybranie celu sygnalizowano poprzez białą obwódkę. Zadania były realizowane w blokach (17 wyborów obiektów). W każdym bloku wszystkie obiekty były tej samej wielkości wizualnej – jeden poziom szerokości celu. Pierwsze dwa cele w każdym bloku były typu statycznego. Pozostałe piętnaście wyborów składało się z pięciu powtórzeń

każdego typu celu, z losowym rozkładem typów celu wokół miejsc na kole obiektów. Uczestnicy wykonywali dziewięć bloków wyborów – początkowy blok przygotowawczy z celami o szerokości 24 pikseli oraz dwa bloki dla każdej szerokości celu w losowej kolejności. Nie uprzedzano użytkowników o typie i charakterze rozszerzania obiektów.

Wyniki badań i komentarze – eksperyment II

W wyniku badań stwierdzono łącznie 28 błędów, poziom błędu równy 1,6%, z tego 10 błędów dla celów statycznych, 12 błędów dla celów z rozszerzeniem wizualnym i 6 błędów dla celów z rozszerzeniem przestrzeni ruchu. Stwierdzono istotny wpływ szerokości celu na poziom błędów, z błędami rosnącymi w miarę zmniejszania się wielkości celu. Dane z badań, które obejmowały błąd, były odrzucane w analizie związanej z czasem wprowadzenia (pozyskania) obiektu [3].

Średni czas pozyskania obiektu dla wszystkich warunków był równy 0,981 s (odch. stand. 0,24 s). Stwierdzono istotny wpływ typu celu na czas pozyskania obiektu, przy czym wartości średnie czasów dla celów statycznych, rozszerzanych wizualnie i z rozszerzaną przestrzenią ruchu były równe 1,033 s (odch. stand. 0,26 s), 0,967 s (odch. stand. 0,20 s) i 0,943 s (odch. stand. 0,24 s) odpowiednio. Stwierdzono również istotny wpływ szerokości celu na czas pozyskania obiektu. Czas działania użytkowników w zależności od typu obiektu scharakteryzowano za pomocą prawa Fittsa (tab. 2).

Tab. 2. Modele prawa Fittsa dla trzech typów obiektów [3]

Typ obiektu	$t_r = a + b\,ID$	R^2
Statyczny	$t_r = 0{,}05 + 0{,}197\,ID$	0,99
Z rozszerzaniem wizualnym	$t_r = 0{,}24 + 0{,}145\,ID$	0,99
Z rozszerzaną przestrzenią ruchu	$t_r = 0{,}13 + 0{,}164\,ID$	0,99

Badania wykazały również, że oba typy rozszerzania celów miały największy pozytywny wpływ (zmniejszały czas wprowadzenia obiektu) dla małych celów. Eksperymenty opisane w artykule sugerują, że w przypadku małych celów znaczną poprawę osiągamy poprzez stosowanie rozszerzania wizualnego bez rozszerzania przestrzeni motorycznej. Artykuł zawiera również komentarze uczestników badania dotyczące używanych form rozszerzania obiektów [3].

Badania zgodne z rekomendacjami normy [6], związane z oceną jakości nowego urządzenia (swiftpoint), przedstawiono w pracy [2]. Swiftpoint to urządzenie stosowane zasadniczo w komputerach mobilnych. Urządzenia tego

można używać na każdej płaskiej powierzchni. Jakość urządzenia porównywano z jakością myszy i touchpada – typowych urządzeń wskazujących również dla komputerów mobilnych. Zaletą swiftpointa jest „zmuszanie" użytkownika do trzymania rąk na płaskiej klawiaturze komputera mobilnego, podczas pisania lub przeglądania. Eliminuje się w ten sposób ruch ręki między klawiaturą i myszą, co jest zgodne z zaleceniami Departamentu Pracy Stanów Zjednoczonych, dotyczącymi bezpieczeństwa pracy [2].

Stosowane urządzenia

Eksperyment przeprowadzono, wykorzystując komputer z procesorem AMD Athlon 64 3200+ z 1 GB pamięci RAM. Grafikę zapewniała karta graficzna GeForce 6600 GT. Wykorzystano 19-calowy monitor Compaq 9500 o rozdzielczości 1600×1200 pikseli (częstotliwość odświeżania monitora 75 Hz). System operacyjny Windows XP. Trzy wejściowe urządzenia wskazujące: mysz – Microsoft IntelliMouse, touchpad – Cirque Smart Cat i Swiftpoint użyty z tabletem Wacom CintiqPartner [2].

Badani użytkownicy

Badanymi osobami było 15 praworęcznych studentów (11 mężczyzn i 4 kobiety), średnia wieku 23 lata. Wszyscy uczestnicy badań używali myszy codziennie.

Procedura badawcza

Podstawą eksperymentu był wielokierunkowy test wskazywania zgodny z normą [6]. Eksperyment składał się z 6 bloków zadań dla każdego urządzenia, w którym blok polegał na kliknięciu w 26 celi. Każdy blok miał inny wskaźnik trudności (3,17; 3,91; 4,75; 5,25; 6,25; 6,98), określony przez dwie średnice koła (300 i 500 pikseli) oraz cztery szerokości celu (4, 8, 19 i 34 piksele).

Obiekty umieszczone na kole były koloru czerwonego, natomiast cel, który należało kliknąć podświetlał się na zielono. Po kliknięciu celu, jego kolor zmieniał się na czerwony, a kolejny cel podświetlał się na zielono. W prawym górnym rogu ekranu wyświetlana była stopa błędu. Podczas badań obowiązywały następujące zasady [2]. Należało klikać w obiekty szybko i dokładnie. Blok zadań powinien być wykonywany w sposób ciągły, przerwę można zrobić po ukończeniu bloku zadań. Podwójne kliknięcie w cel lub kliknięcie poza podświetlonym celem były liczone jako błąd. Przy popełnieniu błędu należy zadanie powtórzyć. W czasie realizacji zadań należy utrzymać stopę błędu na poziomie 4% (należało wolniej i dokładniej wybierać obiekty, jeśli poziom błędów przekraczał 4%).

Wyniki badań i komentarze

Na podstawie wyników badań wyznaczono średni czas wybrania obiektu 1,7 s i odchylenie standardowe 0,68 s. Wyniki szczegółowe dla poszczególnych urządzeń przedstawiono w tab. 3.

Tab. 3. Średni czas wybrania obiektu, odchylenie standardowe i błąd standardowy dla trzech urządzeń [2]

Urządzenie	Średni czas [s]	Odch. stand. [s]	Błąd stand.
Mysz	1,24	0,37	0,04
Touchpad	2,23	0,69	0,07
Swiftpoint	1,61	0,51	0,05

Stwierdzono istotny wpływ rodzaju urządzenia wskazującego na uzyskane wyniki. W tabeli 4 przedstawiono czas ruchu (modele prawa Fittsa por. wzór (6)), współczynnik korelacji R^2 między czasem ruchu i wskaźnikiem trudności ID oraz przepustowość P_w (por. wzór (4)) dla trzech stosowanych urządzeń.

Tab. 4. Modele prawa Fittsa, współczynnik korelacji i przepustowość dla trzech urządzeń [2]

Urządzenie	$t_r = a + b\,ID$	R^2	P_w
Mysz	$t_r = -0{,}01 + 0{,}25\,ID$	0,95	4,05
Touchpad	$t_r = 0{,}11 + 0{,}42\,ID$	0,98	2,38
Swiftpoint	$t_r = -0{,}04 + 0{,}33\,ID$	0,97	3,05

W wynikach badań przedstawiono również szczegółowe wartości średniego poziomu błędów w zależności od wskaźnika trudności. Uczestnicy badań byli bardziej dokładni, gdy używali swiftpointa niż w przypadku używania myszy dla niskich wartości wskaźnika trudności. Dla wyższych wartości wskaźnika trudności uczestnicy badań używający touchpada i swiftpointa koncentrowali się bardziej na szybkości wprowadzania niż na dokładności. Nie dotyczyło to myszy używanej codziennie przez uczestników badań. Dalsze badania przedstawione w pracy [2] dotyczyły wykorzystania tych samych urządzeń w zadaniach nawigacji poprzez zagnieżdżone menu, przeciągania, rysowania i pisania, których większość jest wykonywana regularnie przez użytkowników komputerów podczas interakcji z graficznymi interfejsami użytkownika. Badania te nie będą omawiane, ponieważ zadania te nie dotyczą wielokierunkowego testu wskazywania.

Do okresu badań, w których stosowano klasyczne warunki i typowe urządzenia, można zaliczyć również badania przedstawione w pracy [10]. W badaniach tych badano urządzenia z zastosowaniem zmodyfikowanego testu wielokierunkowego. Badanymi urządzeniami były prototypowe (jak zaznaczono) nastawniki kulowe typu Trackmouse (WingMan, proste urządzenie

z sieci marketów Auchan, TrackMan i Marble Mouse) – dwa ostatnie urządzenia firmy Logitech [10]. Test wielokierunkowy wykorzystano tylko w pierwszym eksperymencie i tylko do wybrania prototypu Trackmouse, który był wykorzystywany w dalszych doświadczeniach. Wykonywano zadania typowe dla testu w normie [6] oraz zadania z dwoma wskaźnikami, używając funkcji myszy i manipulatora kulowego Trackmouse. Zadanie klasyczne (30 obiektów rozłożone na okręgu) realizowano przy średnicy koła równej 22 cm, dla czterech różnych wielkości celów (5, 10, 20 albo 40 mm). Zadanie z dwoma wskaźnikami polegało na wyborze kolejnych obiektów, raz jednym, raz drugim wskaźnikiem, wzdłuż okręgu koła. Nie przesuwano wskaźników przez średnicę koła, ale wymagało to od uczestników badania przenoszenia (i koncentracji) uwagi z jednego na drugi wskaźnik.

Badania w dalszych eksperymentach przedstawionych w pracy [10] dotyczyły zadania zakreślania obiektów (zakreślenie elipsą kropek o określonym kolorze), wyboru obiektów na pasku narzędziowym i obiektów w trzech różnych odległościach z uwzględnieniem treningu w wykorzystaniu urządzeń. Badania te nie będą omawiane, ponieważ zadania te nie dotyczą wielokierunkowego testu wskazywania.

Wielokierunkowy test wskazywania, w zasadzie zgodny z normą ISO 9241-9, był używany w eksperymencie dotyczącym wskazywania obiektów za pomocą zaprojektowanego tzw. bezpośredniego wskaźnika (ang. *Direct Pointer*) na wielkoformatowym zobrazowaniu (70-calowy wskaźnik) [11]. Bezpośredni wskaźnik pozwala na bezpośrednią manipulację wskaźnikiem (kursorem) na ekranie z ciągłym wizualnym sprzężeniem zwrotnym, bardzo przypominając wskaźnik laserowy. Umożliwia on interakcje z dużymi wyświetlaczami intuicyjnie przy użyciu kamery, w które wyposażone są urządzenia przenośne (np. telefony komórkowe i palmtopy). Wyniki badań bezpośredniego wskaźnika porównano z uzyskanymi z literatury wynikami badań innych systemów interakcji, które oceniano na postawie tego samego testu.

3.3. Okres stosowania zmiennych warunków i nietypowych urządzeń

Od roku 2007 zaczynają się pojawiać badania, w których stosowano zmienne warunki badań i nietypowe urządzenia wejściowe. Do takich badań należą badania przedstawione w pracy [25]. Badania te oparte na teście wielokierunkowym [6] dotyczyły śledzenia ruchu oczu (ang. *Eye tracking*) wykonujących zadania wskazywania i wybierania. Oceniano trzy techniki ruchu oczu przy wskazywaniu oraz wybieraniu obiektów i porównywano je z zadaniami realizowanymi za pomocą standardowej myszy.

Stosowane urządzenia

Urządzeniem wejściowym był system śledzenia oczu przy nieruchomej głowie [25]. Stosowano nieruchomą kamerę termowizyjną skierowaną na „dominujące" oko użytkownika. Wykorzystywano 19-calowy monitor LCD o rozdzielczości 1280×1024 pikseli. Uczestnik badań siedział w odległości około 60 cm od ekranu. Urządzenie śledzące oko umożliwiało próbkowanie z częstotliwością 30 Hz z dokładnością od 0,25° do 1,0° kąta widzenia. Przed pierwszym użyciem w czasie badania stosowano kalibrację urządzenia śledzącego.

Badani użytkownicy

Badanymi osobami było szesnastu uczestników (11 mężczyzn, 5 kobiet). Wiek uczestników wahał się od 22 do 33 lat (średnia 25 lat). Wszyscy codziennie korzystali z komputera (4 do 12 h użytkowania dziennie – średnia 7 h). Żaden z uczestników nie miał wcześniejszego doświadczenia ze śledzeniem oczu. Wszyscy uczestnicy mieli normalny wzrok, z wyjątkiem jednego uczestnika, który nosił soczewki kontaktowe. Dziewięciu uczestników miało prawe oko dominujące, siedmiu lewe oko dominujące, co określono za pomocą testu dominacji oka [25].

Procedura badawcza

W badaniach stosowano następujące techniki ruchu oczu przy wskazywaniu oraz wybieraniu obiektów [25]:

- ETL – technika wybór okiem długi (ang. *Eye Tracking Long*), polegająca na spojrzeniu na cel na ekranie i zatrzymaniu na nim wzroku na 750 ms, aby dokonać wyboru;
- ETS – technika wybór okiem krótki (ang. *Eye Tracking Short*), polegająca na spojrzeniu na cel na ekranie i zatrzymaniu na nim wzroku na 500 ms, aby dokonać wyboru;
- ESK – technika oko + wybór spacją (ang. *Eye+Spacebar*), polegająca na spojrzeniu na cel na ekranie i naciśnięciu spacji, aby dokonać wyboru.

Wymienione techniki wyboru porównywano z klasyczną techniką wyboru obiektów za pomocą myszy (M).

Poza 16 obiektami (celami) umieszczonymi na okręgu koła (liczba celi inna niż w normie [6]), w środku koła był umieszczony obiekt (czerwony kwadrat z białym tłem), na którym badana osoba musiała skoncentrować (zogniskować) wzrok przed każdą próbą. Po skoncentrowaniu wzroku znikał kwadrat środkowy i następowała rejestracja czasu. Następne skoncentrowanie wzroku następowało na wybieranym obiekcie (niebieska kropka z niebieskim obrysem). Jeśli wybór obiektu nie nastąpił w ciągu 2,5 s, to rejestrowano błąd związany z czasem (błąd czasu). Potem następowała kolejna próba. Jeśli wzrok

skoncentrowano na celu w czasie ≤ 2,5 s, to wybór był poprawny, czas był zarejestrowany i obraz obiektu zmieniał się na czerwoną kropkę z białym tłem.

W badaniach przyjęto następujące parametry: szerokości celu 75 pikseli i 100 pikseli; odległości obiektów 275 pikseli i 350 pikseli; liczba prób 16.

Wyniki badań i komentarze

Na podstawie wyników badań wyznaczono przepustowość jak w zależności (4). Najlepszą techniką (najwyższa przepustowość) była ESK spośród trzech technik śledzenia oczu (3,78 bitu/s). Pozostałe techniki – ETS 3,06 bitu/s i ETL 2,3 bitu/s. Natomiast mysz, zgodnie z przewidywaniami, miała przepustowość 4,68 bitu/s [25].

Wyznaczono również czas wskazania i wyboru (suma czasu pozycjonowania wzroku i czasu wyboru). Technika ESK była najlepsza pod tym względem, nawet w porównaniu z myszą. Ponadto wszystkie techniki różniły się istotnie pod względem czasu wskazania i wyboru w stosunku do myszy, poza techniką ETS.

Jeśli wziąć pod uwagę poziom błędu i błąd czasu, to dla technik ETL i ETS, uczestnicy wybierali cel przez zatrzymanie wzroku na nim. Wynikiem był zatem albo wybór albo błąd czasu. W związku z tym, poziom błędu dla ETL i ETS był równy zeru, natomiast błąd czasu był równy 19,34% i 14,65% odpowiednio. Błędy te były spowodowane głównie przez fluktuacje oczu i dokładność systemu śledzącego oko. Dla techniki ESK błąd czasu był równy 2,89%, co było znacznie bliższe wartości 1,07% błędu czasu dla myszy, w porównaniu z innymi technikami. Jednakże technika ESK miała wysoki poziom błędu (16,94%). Jest to klasyczny problem kompromisu dokładności i szybkości. Poziom błędu w technice ESK zmieniał się znacznie dla wszystkich uczestników badania (odchylenie standardowe 11,43%). Analiza danych [25] ujawniła, że dla technik ETL, ETS i ESK błąd czasu dla obiektów dużych był niższy niż dla obiektów małych, czego należało się spodziewać.

W pracy [25] przedstawiono również wyniki wypełniania kwestionariusza oceny urządzenia przez uczestników badania (12 pytań). Pytania dotyczyły w ogólności śledzenia okiem w porównaniu do innych technik śledzenia oczu, obciążenia i komfortu użytkowania systemu śledzenia oczu. Uczestnicy badania preferowali technikę ESK spośród stosowanych technik.

Jak zaznaczono w pracy [25], była to pierwsza ocena technik śledzenia oczu zgodna z normą [6]. Badania wykazały, że technika ESK była najlepsza spośród pozostałych badanych technik.

Uzupełnieniem do przedstawionych w pracy [25] badań jest fragment późniejszych badań przedstawiony w publikacji [12]. W badaniach tych oceniano i porównywano czas zatrzymania wzroku w stosunku do mrugnięcia, które pozwalało wybrać cel.

Stosowane urządzenia

Stosowano system śledzenia EyeTech Digital Systems3 TM3 pracujący z oprogramowaniem Quick Glance version 5.0.1. Komputer laptop Lenovo 3000 N100 z ekranem 15-calowym z systemem Microsoft Windows XP. Urządzenie śledzące oko umożliwiało próbkowanie z częstotliwością 30 Hz z polem widzenia 16×12 cm i gęstością pikseli 64,6 piksela/cm [12].

Badani użytkownicy

Badane osoby to 12 uczestników (9 mężczyzn, 3 kobiety). Żaden z uczestników nie miał wcześniejszego doświadczenia ze śledzeniem oczu. Wszyscy uczestnicy mieli normalny wzrok, z wyjątkiem jednego uczestnika, który nosił soczewki kontaktowe.

Procedura badawcza

W badaniach przyjęto następujące parametry [12]:

- metoda wprowadzenia obiektu – mrugnięcie (eye tracker), zatrzymanie wzroku (eye tracker), mysz;
- szerokość celu (średnica) 16 i 32 piksele (6, 12 mm);
- odległość celu – 256, 512 pikseli;
- cztery bloki badań.

Zarówno dla mrugnięcia, jak i zatrzymania wzroku czas był ustawiony na 500 ms. Procedura badania odpowiadająca normie [6] była częściowo podobna do procedury przedstawionej w pracy [25]. Różnice dotyczyły braku obiektu w centrum koła z celami i podświetlania na czerwono obiektu wybieranego.

Wyniki badań i komentarze

Mysz miała znacznie większą przepustowość niż metody wykorzystujące system śledzenia oka (eye tracking). Przepustowość dla myszy była równa 4,79 bitu/s, wartość ta jest bardzo bliska wartości 4,68 bitu/s z poprzedniego badania [25] (przy użyciu innego komputera i innego oprogramowania). Wynik ten świadczy o stałości zasad normy ISO 9241-9 w ocenie urządzeń wskazujących. Dla pozostałych metod przepustowość była równa 1,79 bitu/s dla metody z zatrzymaniem wzroku i 1,16 bitu/s dla metody wyboru z mrugnięciem.

Biorąc pod uwagę niewielkie cele w tym badaniu, poziomy błędów były wysokie (czego można było się spodziewać) – sięgały 50-80% dla metod wykorzystujących system śledzenia oka (w zależności od bloku badań). Uwagi związane z małą dokładnością wyboru obiektów dotyczą głównie małej wielkości celów na ekranie. Wymiary obiektów 32 i 16 pikseli, czyli około 12 mm i 6 mm w połączeniu z odległością widzenia około 60 cm, dają mały kąt widzenia 1,15° i 0,57° odpowiednio.

Badania przedstawione w pracy [18] dotyczyły wpływu opóźnienia oraz przestrzennych fluktuacji urządzenia wejściowego na szybkość wskazywania obiektów i dokładność. Porównywano przepustowość dla różnych wielkości opóźnienia pomiędzy ruchem urządzenia i ruchem wskaźnika na ekranie oraz dla różnych wielkości drgań przestrzennych wskaźnika. Badania wykorzystywały test wielokierunkowy przedstawiony w normie [6].

Stosowane urządzenia

W badaniach wykorzystano komputer stacjonarny oparty na Intel Pentium 4, 2,4 GHz, z pamięcią 1 GB RAM. Urządzeniem wejściowym była mysz optyczna Microsoft Wheel.

Badani użytkownicy

Uczestnikami badania było dwanaście osób (w tym ośmiu mężczyzn) w wieku od 19 do 31 lat (średnia wieku 23 lata). Wszyscy byli praworęczni. Badanie trwało 30-40 minut [18].

Procedura badawcza

Test wielokierunkowy stosowany w badaniach zawierał 13 obiektów (celów) umieszczonych na okręgu koła (liczba celi inna niż w normie [6]). Badanie przebiegało wg klasycznej procedury (cele wybierane naprzeciwko, podświetlanie celu). Oprogramowanie rejestrowało rozmiary celów, odległości między celami, czasy kliknąć między celami, błędy i współrzędne ekranowe każdego kliknięcia. Uczestnicy badania siedzieli przed ekranem komputera, w odległości około 0,6 m. Przed badaniem uczestnicy zostali zapoznani z systemem i mogli go wypróbować.

W badaniach stosowano następujące wielkości zmieniane:

- opóźnienie 33, 58, 83, 108 i 133 ms;
- drgania (fluktuacje) przestrzenne 0, ± 4, ± 8, ± 12 i ± 16 pikseli;
- szerokość celu (średnica) 14, 35 i 91 pikseli;
- amplitudy celu (średnica okręgu) 416 i 728 pikseli.

Połączenie szerokości celów i średnicy okręgu tworzy równomierne odstępy wartości wskaźnika trudności od 2,5 do 5,7 bitu. Każdy uczestnik wykonywał 150 rund z różnymi wartościami parametrów [18].

Wyniki badań i komentarze

Wyznaczono następujące wielkości [18]:

- przepustowość urządzenia (bit/s), obliczone jak w zależności (4);
- procent celów nie trafionych (procent błędów, poziom błędów) w każdej rundzie.

Sprawdzono, że istotny był wpływ opóźnienia na przepustowość. Analizując wyniki przedstawione na wykresie, można wywnioskować, że im

większe opóźnienie, tym mniejsza wartość przepustowości. Związek między wielkością opóźnienia i szerokością obiektu miał istotny wpływ na przepustowość. Dla każdej wartości opóźnienia – im mniejszy obiekt, tym mniejsza przepustowość.

Sprawdzono, że wpływ drgań (fluktuacji) na przepustowość był istotny. Analizując wyniki, można wywnioskować, że im większe drgania, tym mniejsza wartość przepustowości. Związek między wielkością drgań i szerokością obiektu miał istotny wpływ na przepustowość. Dla każdej wielkości drgań – im mniejszy obiekt, tym mniejsza przepustowość.

Wpływ szerokości celu na przepustowość był również istotny (większa szerokość celu – wyższa przepustowość). Podobnie wpływ amplitudy celu (średnicy okręgu) był istotny (3,93 bitu/s dla 416 pikseli w porównaniu do 4,09 dla 728 pikseli). Wpływ wskaźnika trudności na przepustowość był również istotny.

Sprawdzono, że istotny był wpływ opóźnienia na poziom błędów. Związek między wielkością opóźnienia i szerokością obiektu miał istotny wpływ na poziom błędów. Stwierdzono również, że wpływ wielkości drgań (fluktuacji) na poziom błędów był istotny. Związek między wielkością drgań i szerokością obiektu miał istotny wpływ na poziom błędów.

Wpływ szerokości celu na poziom błędów był również istotny (większa szerokość celu – niższy poziom błędów). Nie stwierdzono statystycznej istotności wpływu odległości celów na poziom błędu.

Stwierdzono, że istotny był wpływ opóźnienia na czas przemieszczania wskaźnika na ekranie. Związek między wielkością opóźnienia i szerokością obiektu miał istotny wpływ na czas przemieszczania wskaźnika na ekranie. Zarówno amplituda celu (średnica okręgu), jak i szerokość celu miały istotny wpływ na czas przemieszczania wskaźnika na ekranie (większa szerokość celu – krótszy czas przemieszczania wskaźnika).

Szczegółowe wyniki badań przedstawione są na wykresach [18]. Ogólnie patrząc na wyniki, można zaobserwować, że wydajność spada ze wzrostem opóźnień i drgań przestrzennych. Wzrasta również poziom błędów. Należy zwrócić uwagę, że w badaniach nie wyznaczono wielkości stałych występujących w równaniu prawa Fittsa (por. wzór (6)).

Różnicowanie (porównywanie) urządzeń z zastosowaniem testu wielokierunkowego przedstawiono w pracy [17]. Badania porównawcze dotyczyły porównania prototypu kontrolera gry do standardowego kontrolera gier. W prototypowym kontrolerze zastąpiono trackballem prawy analogowy drążek standardowego kontrolera gier (używanego do wskazywania i sterowania kamerą).

Stosowane urządzenia

Używano projektora DLP NEC NP60 (rozdzielczość 1024 × 768) w celu symulowania ekranu wielkoformatowego, ponieważ użytkownicy zwykle używają konsoli do gier na dużym ekranie telewizyjnym [11]. Zastosowano komputer z systemem Windows XP. Uczestnicy badań siedzieli 3 metry od wyświetlanego obrazu. Przekątna wyświetlanego obrazu była równa 115 cm.

Prototyp kontrolera trackballa został zbudowany z przewodowego kontrolera Microsoft Xbox 360 oraz trackballa Logitech Trackman Wheel. W prototypie pokrętło zastąpiono trackballem [17]. Użyto oprogramowania, które umożliwia emulację wejścia kontrolera gier.

Badani użytkownicy

Badaniami objęto dwie grupy uczestników, początkujących graczy (5 kobiet i 5 mężczyzn) i zaawansowanych użytkowników kontrolera (10 mężczyzn). W kwestionariuszu pytano uczestników o doświadczenie w używaniu standardowych kontrolerów gier.

Początkujący użytkownicy – średnia wieku wynosiła 25,3 (odch. stand. 3,19). Zaawansowani użytkownicy – średnia wieku wynosiła 25,2 (odch. stand. 3,96).

Procedura badawcza

W eksperymencie zastosowano test wielokierunkowy [6]. Siedemnaście okrągłych celów było rozłożonych na wyśrodkowanym na ekranie okręgu. Cele były umieszczone w regularnych odstępach wzdłuż obwodu koła (zgodnie z normą ISO 9241-9). Następny cel do kliknięcia był podświetlony na czerwono. Kliknięcie w pierwszy aktywny cel zaczynało próbę i przeciwległy cel stawał się aktywny. Przy braku trafienia w cel pojawiał się sygnał dźwiękowy.

Po zakończeniu wszystkich prób dla danego okręgu, pojawiało się podsumowanie wyników uczestnika. Po akceptacji pojawiał się następny okrąg z obiektami dla kolejnego wskaźnika trudności ID.

W badaniach stosowano następujące wielkości zmieniane [17]:

- doświadczenie (uczestnicy początkujący, zaawansowani);
- rodzaj kontrolera (standardowy, z trackballem);
- szerokość celu (średnica) 20, 35 pikseli;
- odległość celów (średnica okręgu) 128, 256 i 512 pikseli;
- siedem bloków zadań.

Rejestrowano czas między kliknięciami i fakt błędnego wyboru obiektu. Uczestnicy badania byli wcześniej instruowani o zadaniach i wykonywali pojedyncze badanie treningowe.

Wyniki badań i komentarze

Wyniki dla początkujących uczestników. Średnia przepustowość standardowego kontrolera dla wszystkich siedmiu bloków była równa 1,68 bitu/s (odch. stand. 0,07). Przepustowość kontrolera z trackballem była równa 2,69 bitu/s (odch. stand. 0,14). Różnice te były istotne statystycznie.

Wyniki dla zaawansowanych uczestników. Średnia przepustowość standardowego kontrolera dla wszystkich siedmiu bloków była równa 2,01 bitu/s (odch. stand. 0,08). Przepustowość kontrolera z trackballem była równa 3,19 bitu/s (odch. stand. 0,19). Różnice te były istotne statystycznie.

Dla początkujących użytkowników i dla standardowego kontrolera, średni poziom błędu był równy 5,81% (odch. stand. 0,57%) i 2,96% (odch. stand. 0,44%) dla kontrolera z trackballem. Różnica ta nie była statystycznie istotna. Dla grupy zaawansowanej i dla standardowego kontrolera, średnia stopa błędu była równa 5,87% (odch. stand. 0,80%) i 5,63% (odch. stand. 0,91%) dla kontrolera z trackballem. Wyniki te również nie były statystycznie istotne.

Wyznaczano również tzw. błąd ruchu [17] – miarę obliczaną na podstawie współrzędnych ścieżki ruchu wskaźnika, oznaczającą średnią odchylenia punktów od osi zadania (miara bezwzględna przy przyjęciu, że oś zadania ma $y = 0$). Wyniki te wskazują, że ścieżka wskaźnika jest gładsza dla standardowego kontrolera. Pomimo tego można stwierdzić, że kontroler z trackballem oferuje znaczną poprawę wydajności, zarówno dla początkujących, jak i zaawansowanych użytkowników, w stosunku do standardowego kontrolera.

Na wykresach przedstawiono szczegółowe wyniki uzyskane dla każdego bloku zadań. Pokazano również typowe ścieżki ruchu wskaźnika między wybieranymi obiektami dla obu kontrolerów [17].

Badania nietypowej metody wprowadzania przedstawiono w pracy [16]. Oceniano pochylenie jako metodę wprowadzania dla urządzeń mobilnych z wbudowanym akcelerometrem, takich jak telefony z ekranem dotykowym i tablety. Badano, w jakim stopniu użytkownicy mogą sterować obiektem na ekranie przy użyciu pochylenia urządzenia oraz jakie parametry pochylenia i w jaki sposób wpływają na wydajność użytkowników.

Stosowane urządzenia

Badania przeprowadzono, używając tabletu Samsung Galaxy Tab 10.1 z systemem operacyjnym Android 3.1. Sterowanie pochyleniem zostało zrealizowane za pomocą wbudowanego akcelerometru urządzenia. Czujnik został skonfigurowany do działania z częstotliwością próbkowania 50 Hz. Wartości pochylenia i obrotu były przeliczane na wielkość przechylenia i kąt przechylenia. Dane te wykorzystywano do sterowania kierunkiem i szybkością wirtualnego toczenia kulki w interfejsie. Szybkość kulki była liniową funkcją wielkości pochylenia [16].

Badani użytkownicy

Badanymi użytkownikami było 16 osób (9 mężczyzn i 7 kobiet), w wieku od 19 do 39 lat (średnia = 26 lat, odch. stand. 5,0) [16]. Większość uczestników badania zgłaszała niewielką znajomość interakcji opartych na nachyleniu. Sześć osób nigdy go nie używało i osiem używało go jedynie sporadycznie (kilka razy w miesiącu). Pozostali dwaj uczestnicy stosowali interakcje oparte na nachyleniu częściej.

Procedura badawcza

W badaniach stosowano test wielokierunkowy [6]. Zadaniem użytkowników było przechylenie urządzenia w celu sterowania kierunkiem i szybkością wirtualnej kulki. Zmierzone pochylenie było zmieniane (mnożone) przez wzmocnienie możliwe do ustawienia. Kulkę należało przenieść z jednego kółka (celu) do drugiego. Na okręgu było umieszczonych dwanaście kółek (celów). Wirtualna kulka miała średnicę 20 pikseli. W badaniach użyto trzech rozmiarów kółek (celów) 40, 60 i 100 pikseli. Zastosowano trzy średnice okręgu, na którym były umieszczone cele 125, 250 i 500 pikseli (500 pikseli = 8,5 cm na ekranie). W tej sytuacji wskaźnik trudności zadania wahał się od 1,36 bitu do 4,7 bitu.

Realizowano dwa tryby wyboru obiektu na okręgu. Jednym z trybów było „pierwsze wprowadzenie", gdzie próba kończy się, gdy kulka pierwszy raz wchodzi w kółko celu. Drugi z trybów to „utrzymywanie kulki", gdzie próba kończy się po tym, jak kulka jest utrzymywana wewnątrz kółka celu w określonym przedziale czasu. Dla tego trybu należało utrzymywać kulkę w celu przez 500 ms. Oba tryby wyboru wymagały, aby kulka całkowicie zmieściła się w kółku celu. Podczas eksperymentu uczestnicy siedzieli i trzymali tablet w komfortowej dla siebie pozycji. Uczestnicy mogli wykonać kilka prób treningowych przed rozpoczęciem zbierania danych. Blok badań składał się z dziewięciu sekwencji (3 średnice okręgu × 3 szerokości celów) prezentowanych w losowej kolejności. Sekwencja składała się z dwunastu wyborów celów. Po badaniach pilotażowych do badania użytkowników zostały wybrane cztery ustawienia wzmocnienia pochylenia 25, 50, 100 i 200 [16].

Wyniki badań i komentarze

Wyniki badań dotyczące czasu ruchu kulki były następujące. Średni czas ruchu dla całego eksperymentu był równy 2026 ms. Zgodnie z oczekiwaniami, wpływ trybu wyboru obiektu na okręgu na czas ruchu był statystycznie istotny. Średni czas dla trybu pierwsze wprowadzenie był równy 1404 ms, natomiast dla trybu utrzymywanie kulki wewnątrz kółka celu był równy 2647 ms. Czas ten jest znacznie większy niż naturalna różnica 500 ms i uzasadniono to problemami ze sterowaniem kulką przy wyższym wzmocnieniu pochylenia.

Wpływ wzmocnienia pochylenia na czas ruchu kulki był również istotny statystycznie. Przy wzmocnieniu 50 uzyskano najkrótszy czas ruchu (1900 ms), przy wzmocnieniu pochylenia 200 najdłuższy czas (2210 ms) [16].

Jako miarę dokładności podano średnią zmienność od bezpośredniej ścieżki w każdych warunkach (była ona między 7 i 19 pikseli). Tryb wyboru obiektu i wzmocnienie pochylenia były istotne statystycznie. W wynikach badań podano również parametry modelu czasu ruchu (modele prawa Fittsa por. wzór (6)), współczynnik korelacji R^2 między czasem ruchu i wskaźnikiem trudności ID, dla różnych wartości wzmocnienia pochylenia (tab. 5).

Tab. 5. Modele prawa Fittsa i współczynnik korelacji dla różnych trybów wyboru obiektu i różnych wartości wzmocnienia pochylenia [16]

Wzmocnienie pochylenia	Pierwsze wprowadzenie			Utrzymywanie kulki 500		
	a	b	R^2	a	b	R^2
25	38,6	486	0,9654	222,9	789	0,9923
50	91,6	427	0,9829	91,7	814	0,9810
100	11,1	473	0,9648	13,5	894	0,9668
200	7,6	491	0,9443	-270,6	1116	0,8841

Wyniki wykazały, że wzmocnienie pochylenia w zakresie od 50 do 100 jest optymalne dla czasu ruchu kulki i przepustowości. Wybór obiektu w trybie pierwsze wprowadzenie jest szybszy niż utrzymanie kulki i powinien być stosowany w aplikacjach. Badania wykazały, że pochylenie jako wejście prymitywne – jest zgodne z prawem Fittsa, chociaż wydajność jest niższa o połowę w stosunku do myszy komputerowej [16].

W interfejsach typu wskaż i kliknij lokalizacja celów jest czasem znana użytkownikowi przed jej identyfikacją wizualną, a czasem nie. W eksperymencie przedstawionym w pracy [8] badano, jak wskazywanie obiektów jest skuteczne, gdy lokalizacja celu jest wskazana wcześniej tak, że użytkownicy wiedzą z góry, gdzie jest cel lub nie jest znane wcześniej położenie celu.

Stosowane urządzenia

Badania prowadzono na laptopie HP 1,86 GHz z myszą przewodową, wbudowanym touchpadem typu Synaptics Touchpad o wymiarach 68 mm × 39 mm i 15-calowym ekranem o rozdzielczości 1024×768 pikseli. Mysz miała dwa przyciski, ale tylko lewego przycisku używano w eksperymencie. Wybór za pomocą touchpada wykonywano poprzez puknięcie w powierzchnię lub wciśnięcie lewego przycisku [8].

Badani użytkownicy

Badano 36 osób – 18 kobiet i 18 mężczyzn w wieku od 12 do 69 lat (średnia wieku 34,75, odch. stand. 21,17). Wszyscy byli praworęczni i mieli

doświadczenie w użyciu myszy i touchpada. Uczestnicy badań byli podzieleni na 3 grupy wiekowe po 12 osób. Grupa pierwsza (młodzi) – osoby w wieku od 12 do 14 lat. Grupa druga (dorośli) – osoby w wieku od 25 do 33 lat. Grupa trzecia (starsi) – osoby w wieku od 61 do 69 lat. Wszyscy uczestnicy wykorzystywali w badaniach zarówno mysz, jak i touchpad. Połowa uczestników w każdej grupie wiekowej stosowała mysz na pierwszą połowę sesji i touchpada w drugiej połowie sesji. Druga połowa osób w grupie wiekowej stosowała touchpad, następnie mysz.

Procedura badawcza

W zadaniach eksperymentalnych osiem obiektów było rozmieszczonych na okręgu wokół obiektu centralnego, a uczestnicy byli zobowiązani do wybierania na przemian obiektu w centrum i jednego z ośmiu otaczających obiektów. Cel, który uczestnik powinien wybrać jako następny, był podświetlony na czerwono, podczas gdy inne obiekty były jasnoniebieskie, wszystkie obiekty na czarnym tle. Wybrany cel wracał do koloru jasnoniebieskiego, a kolejny cel stawał się czerwony. Pierwszym celem w każdym zadaniu był obiekt w centrum. Jego wybór zapoczątkowywał zadania [8].

Uczestnicy nie wiedzieli, który z ośmiu otaczających obiektów będzie do wyboru, ponieważ porządek wyboru celu był losowy (co drugi wybór). Natomiast położenie obiektu centralnego było znane. Wskazanie celu odbywało się przez stałą lokalizację obiektu centrum i systematycznie poprzez wybranie obiektu naprzeciw po wybraniu obiektu wskazanego losowo.

W badaniach stosowano następujące wielkości zmieniane [8]:

- wskazanie celu (ze wskazaniem i bez wskazania – dwa poziomy);
- odległość do celu (70 pikseli (mały), 175 pikseli (średni) lub 350 pikseli (duży));
- wielkość celu (małe cele średnica 6 pikseli, duże cele średnica 21 pikseli);
- rodzaj urządzenia (mysz, touchpad).

Każde zadanie składało się z 32 prób i obejmowało jeden poziom odległości do celu, jeden poziom wielkości celu i oba poziomy wskazania celu. Próby w zadaniach były na przemian ze wskazaniem i bez wskazania celu [8].

Wyniki badań i komentarze

Analizowano tylko wyniki uzyskane z drugiego bloku, żeby uniknąć efektów uczenia się. Odrzucono również 1,5% wszystkich wyników ze względu na dużą rozbieżność

Wyniki badań dotyczące poziomu błędów podano w tab. 6. Wykazano, że nie ma istotnych związków między wskazaniem celu oraz dowolnym urządzeniem wskazującym, odległością do celu i wielkością celu.

Tab. 6. Procent [%] błędów dla badanych urządzeń, odległości do celu i wielkości celu dla celi ze wskazaniem i bez wskazania [8]

	Cele ze wskazaniem		Cele bez wskazania	
	Średnia	Odch. stand.	Średnia	Odch. stand.
Urządzenie				
Mysz	8,3	0,9	8,2	1,0
Touchpad	10,8	1,2	9,2	1,1
Odległość do celu				
70 pikseli	8,8	1,0	9,2	1,1
175 pikseli	9,9	1,1	9,5	0,9
350 pikseli	10,1	1,0	10,6	1,0
Wielkość celu				
6 pikseli	13,5	1,4	14,1	1,3
21 pikseli	5,6	0,6	5,5	0,7
Całkowity	9,6	0,9	9,8	0,9

Wyniki badań dotyczące czasów wykonania prób podano w tab. 6. Wykazano, że są istotne związki między wskazaniem celu oraz odległością do celu i wielkością celu.

Tab. 7. Czasy [ms] wykonania prób dla badanych urządzeń, odległości do celu i wielkości celu dla celi ze wskazaniem i bez wskazania [8]

	Cele ze wskazaniem		Cele bez wskazania	
	Średnia	Odch. stand.	Średnia	Odch. stand.
Urządzenie				
Mysz	1199	25	1393	29
Touchpad	2172	65	2324	66
Odległość do celu				
70 pikseli	1438	38	1596	40
175 pikseli	1702	44	1844	39
350 pikseli	1917	44	2135	51
Wielkość celu				
6 pikseli	2017	50	2221	54
21 pikseli	1381	33	1536	35
Całkowity	1686	41	1858	42

Różnice w średnim czasie realizacji próby między celami ze wskazaniem i celami bez wskazania były większe dla dużej odległości do celu (218 ms), niż dla małych i średnich odległości do celu (158 ms i 142 ms, odpowiednio). Również różnice w średnim czasie realizacji próby między celami ze wskazaniem i celami bez wskazania były większe dla małych celów (204 ms) niż większych celów (155 ms).

W wynikach badań [8] podano również parametry modelu czasu ruchu (modele prawa Fittsa por. wzór (6)), współczynnik korelacji R^2 między czasem ruchu i wskaźnikiem trudności ID, dla celów ze wskazaniem i celów bez wskazania (tab. 8).

Tab. 8. Modele prawa Fittsa i współczynnik korelacji dla celów ze wskazaniem i bez wskazania dla badanych urządzeń [8]

Urządzenie	Cele	$t_r = a + b\,ID$ [ms]	R^2
Mysz	Ze wskazaniem	$t_r = 224 + 250\,ID$	0,93
	Bez wskazania	$t_r = 243 + 293\,ID$	0,93
	Ogółem	$t_r = 233 + 272\,ID$	0,86
Touchpad	Ze wskazaniem	$t_r = 756 + 358\,ID$	0,92
	Bez wskazania	$t_r = 913 + 355\,ID$	0,93
	Ogółem	$t_r = 835 + 357\,ID$	0,90

Opisywane badania [8] wykazały, że wskazanie celu wpłynęło zaskakująco niewiele na wyniki grup wiekowych. Nie stwierdzono istotnych związków pomiędzy wskazaniem celu i wiekiem grupy dla każdego z poziomu błędu i czasu realizacji próby. Wyniki badań zawierają również czasy reakcji, czasy ruchu i czasy wyboru dla celów ze wskazaniem i celów bez wskazania. Czasy te w sumie tworzą czas próby.

Przedstawione wyżej omówienia prowadzonych w ostatnich latach badań nie byłyby pełne, gdyby nie zasygnalizować badań, w których test wielokierunkowy [6] był wykorzystywany w dość specyficznych warunkach.

W badaniach przedstawionych w pracy [23] badano działania osób upośledzonych motorycznie (np. choroba Parkinsona, dystrofia mięśniowa) wykorzystujących tzw. mysz kątową (ang. *Angle mouse*). W badaniach wykorzystano wielokierunkowy test wskazywania [6]. Sprawdzano, czy mysz kątowa może poprawić wydajność wskazywania obiektów w porównaniu do standardowej myszy systemu Windows i tzw. lepiących się ikon (przyciągające ikony), ale bez zmian dla użytkowników pełnosprawnych. Badania wykazały, że

mysz kątowa umożliwia uzyskanie wyższej przepustowości niż domyślna mysz Windows i lepiące się ikony.

Językowy system kierowania (ang. *Tongue Drive System* – TDS) to mobilna technika bezprzewodowa, która umożliwia osobom z poważnymi uszkodzeniami motorycznymi dostęp do komputerów, sterowanie wózkiem inwalidzkim itp., wykorzystując ruch języka. W badaniach przedstawionych w pracy [24] oceniano takie urządzenie wejściowe komputera w różnych zadaniach, z których jednym był test wielokierunkowy [6]. System kierowania TDS wykorzystywał piercing języka magnesem i odpowiednie sensory nagłowne. Badania obejmowały również wprowadzanie poleceń palcem na ograniczonej liczbie klawiszy klawiatury. Wyniki badań to przepustowość, stopa błędów, czas realizacji zadania i tzw. efektywność ścieżki wskaźnika.

Badaniami związanymi z wielokierunkowym testem wskazywania są badania przedstawione w pracy [21]. Badano funkcje nowego bezprzewodowego prototypu interfejsu nazwanego Face Interface. Prototyp łączy wykorzystanie kierunku spojrzenia (eye tracking) i aktywacji mięśni twarzy dla odpowiedniego wskazywania i wybierania obiektów na ekranie komputera. Używano marszczenia brwi i podnoszenia brwi, jako metod wyboru celu. Cele wybierane o trzech średnicach pokazywane były na ekranie na siedmiu różnych odległościach i w ośmiu kierunkach (0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°, i 360°). Wyniki pokazały, że metoda podnoszenia brwi była szybszą metodą wyboru obiektów niż metoda marszczenia brwi dla znacznego zakresu odległości pojawiających się obiektów.

4. Podsumowanie

W artykule przedstawiono wielokierunkowy test wskazywania stosowany w badaniach jakości wskazujących urządzeń wprowadzania. Przedstawiono również wybrane badania opisywane w literaturze, wykorzystujące ten test do oceny jakości działania urządzeń. Badania obejmują publikacje z ostatnich lat.

Zestawienie uzyskanych wyników z badań dla myszy i touchpada przedstawiono w tab. 9. Porównanie wyników dla innych urządzeń nie jest możliwe, ponieważ badania wykonywano raz dla konkretnego urządzenia, którego w późniejszym okresie nie badano. Z przedstawionego zestawienia można by wnioskować o częściowej zgodności wartości wybranych parametrów, w szczególności w zakresie czasu wprowadzenia obiektu (celu) dla touchpada lub przepustowości dla myszy. Należy jednak zwrócić uwagę, że badania wykonywano z wykorzystaniem różnych urządzeń i w różnych warunkach.

Tab. 9. Zestawienie wyników badań dla wybranych dwóch badanych urządzeń

Urządzenie	Stopa błędów [%]	Czas wprow. [s]	Odch. stand. [s]	Przepustowość [bit/s]
Mysz [2]	-	1,24	0,37	-
Mysz [15]	9,4	-	-	4,9
Mysz [25]	1,07	-	-	4,68
Mysz [12]	1,1	-	-	4,79
Mysz [8]	8,3	1,199	-	-
Touchpad [5]	5,4	2,382	0,802	1,70
Touchpad [15]	7,0	-	-	2,9
Touchpad [2]	-	2,23	0,69	-
Touchpad [8]	10,8	2,172	-	-

Opisywane badania realizowano z wykorzystaniem testu wielokierunkowego i często sam test zawierał zmiany w stosunku do klasycznych warunków podanych w normie [6]. Duża część badań opisanych w literaturze dotyczyła realizacji zadań wyboru obiektu poprzez gesty (np. [11], [12], [16], [21], [25]) lub nietypowo język [24]. W badaniach brały udział niezbyt duże grupy użytkowników (od 6 do 36 osób), przy czym tylko w dwóch z przedstawionych badań liczba uczestników przekroczyła 20.

Przegląd badań pozwala na szersze spojrzenie na wyniki wykorzystania testu wielokierunkowego, biorąc pod uwagę stosowane urządzenia wejściowe i warunki, w jakich badania prowadzono.

Literatura:

[1] ACCOT J., ZHAI S., *Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks*, Proceedings of ACM CHI, Conference on Human Factors in Computing Systems, 1997, pp. 295-302.

[2] AMER T., COCKBURN A., GREEN R., ODGERS G., *Evaluating Swiftpoint as a Mobile Device for Direct Manipulation Input,* AUIC2007 (CRPIT), Vol. 64, 2007, pp. 63-70.

[3] COCKBURN A., BROCK P., *Human On-Line Response to Visual and Motor Target Expansion*, Graphic Interface, 2006, pp. 81-87.

[4] DONIGIEWICZ A.M., *Jednokierunkowy test wskazywania – norma ISO 9241-9 – przegląd badań,* Biuletyn IAiR, 30, 2011, s. 71-88.

[5] DOUGLAS A.S., KIRKPATRICK A.E., MACKENZIE I.S., *Testing Pointing Device Performance and User Assessment with the IS0 9241, Part 9 Standard*, CHI '99 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Pittsburgh, 1999, pp. 215-222.

[6] *Ergonomic requirements for Office work with visual display terminals (VDTs). Part 9: Requirements for non-keyboard input devices*, ISO 9241-9:2000 (E), International Organization for Standardization.

[7] FITTS P.M., *The information capacity of the human motor system in controlling the amplitude of movement*, Journal of Experimental Psychology, Vol. 47, No. 6, 1954, pp. 381-391. (Reprinted in Journal of Experimental Psychology: General, 121(3), 1992, pp. 262-269).

[8] HERTZUM M., HORNBÆK K., *The Effect of Target Precuing on Pointing with Mouse and Touchpad*, International Journal of Human-Computer Interaction, Vol. 29, No. 5, 2013, pp. 338-350.

[9] HERTZUM M., HORNBÆK K., *TouchGrid: Touchpad pointing by recursively mapping taps to smaller display regions*, Behaviour & Information Technology, Vol. 24, No. 5, 2005, pp. 337-346.

[10] ISOKOSKI P., RAISAMO R., MARTIN B., EVREINOV G., *User performance with trackball-mice*, Interacting with Computers 19, 2007, pp. 407-427.

[11] JIANG H., OFEK E., MORAVEJI N., SHI Y., *Direct Pointer: Direct Manipulation for Large-Display Interaction using Handheld Cameras*, CHI 2006 Proceedings, Montreal, pp. 1107-1110.

[12] MACKENZIE S.I., *Evaluating eye tracking systems for computer input*, [In:] *Gaze interaction and applications of eye tracking: Advances in assistive technologies*, Hershey, PA: IGI Global, 2012, pp. 205-225.

[13] MACKENZIE I.S., *Motor behaviour models for human-computer interaction,* [In:] *HCI models, theories, and frameworks: Toward a multidisciplinary science*, San Francisco, Kaufmann, 2003, pp. 27-54.

[14] MACKENZIE I.S., *Movement time prediction in human-computer interfaces*, [In:] *Readings in human-computer interaction*, Los Altos, Kaufmann, 1995, pp. 483-493.

[15] MACKENZIE S., KAUPPINEN T., SILFVERBERG M., *Accuracy Measures for Evaluating Computer Pointing Devices*, SIGCHI '01, March 31–April 4, 2001, pp. 9-16.

[16] MACKENZIE I.S., TEATHER R.J., *FittsTilt: The Application of Fitts' Law To Tilt-based Interaction*, NordiCHI '12, October 14–17, 2012, pp. 568-577.

[17] NATAPOV D., MACKENZIE S.I., *The Trackball Controller: Improving the Analog Stick*, Futureplay '10 Proceedings of the International Academic Conference on the Future of Game Design and Technology, ACM, New York, 2010, pp. 175-182.

[18] PAVLOVYCH A., STUERZLINGER W., *The Tradeoff between Spatial Jitter and Latency in Pointing Tasks*, EICS '09, July15–17, 2009, Pittsburgh, pp. 187-196.

[19] SIKORSKI M., *Interakcja człowiek-komputer*, Wyd. PJWSTK, Warszawa, 2010.

[20] SOUKOREFF R.W., MACKENZIE I.S., *Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI*, Int. J. Human-Computer Studies, Vol. 61, 2004, pp. 751-789.

[21] TUISKU O., SURAKKA V., VANHALA T., RANTANEN V., LEKKALA J., *Wireless Face Interface: Using voluntary gaze direction and facial muscle activations for human-computer interaction*, Interacting with Computers, Vol. 24, Issue 1, January, 2012, pp. 1-9.

[22] WHISENAND T.G., EMURIAN H.H., *Effects of Angle of Approach on Cursor Movement with a Mouse: Cosideration of Fitts' Law*, Computers in Human Behaviour, Vol. 12, No. 3, 1996, pp. 481-495.

[23] WOBBROCK J.O., FOGARTY J., LIU S., KIMURO S., HARADA S., *The Angle Mouse: Target-Agnostic Dynamic Gain Adjustment Based on Angular Deviation*, CHI 2009, April 4–9, 2009, Boston, pp. 1401-1410.

[24] YOUSEFI B., HUO X., VELEDAR E., GHOVANLOO M., *Quantitative and Comparative Assessment of Learning in a Tongue-Operated Computer Input Device*, IEEE Transactions on Information Technology in Biomedicine, 2011, pp. 747-757.

[25] ZHANG X., MACKENZIE I.S., *Evaluating Eye Tracking with ISO 9241 – Part 9*, Human-Computer Interaction, Part III, HCII 2007, LNCS 4552, pp. 779-788.

Źródła elektroniczne

[26] BUXTON W., *Theories, models and basic concepts,* [In:] *Haptic Input*, pp. 7.1-7.46 http://www.billbuxton.com/input07.TheoriesModels.pdf (dostęp 20.01.2011).

[27] MACKENZIE I.S., *Fitts' law as a performance model in human-computer interaction,* Unpublished Doctoral Dissertation, University of Toronto http://www.yorku.ca/mack/phd.html (dostęp 20.01.2011).

Multidirectional tapping test – ISO 9241-9 standard – a survey

ABSTRACT: A multidirectional tapping test, applied to the quality assessment of entering information with pointing devices, is presented. The description of the test is based on the ISO 9241-9 standard. The test may be the basis for an evaluation or estimation of the quality of entering information by a user. Some research results available in the bibliography are quoted.

KEYWORDS: tests of non-keyboard input devices, multidirectional tapping test, Fitts' law, ISO 9241-9 standard

Praca wpłynęła do redakcji: 17.12.2014 r.

Recenzenci artykułów czasopisma naukowego
PRZEGLĄD TELEINFORMATYCZNY

Lata 2013-2014

Afonso Joao	Foundation for National Scientific Computing, Portugal
Amanowicz Marek	Wydział Elektroniki, Wojskowa Akademia Techniczna
Barczak Andrzej	Instytut Informatyki, Uniwersytet Przyrodniczo-Humanistyczny w Siedlcach
Bednarczyk Mariusz	Wydział Elektroniki, Wojskowa Akademia Techniczna
Brudka Marek	FILBICO
Cieciura Marek	Wydział Informatyki, Wyższa Szkoła Technologii Informatycznych
Dołowski Jerzy	Wydział Elektroniki, Wojskowa Akademia Techniczna
Furtak Janusz	Wydział Cybernetyki, Wojskowa Akademia Techniczna
Gajewski Piotr	Wydział Elektroniki, Wojskowa Akademia Techniczna
García-Osorio César	University of Burgos, Hiszpania
Gniazdowski Zenon	Warszawska Wyższa Szkoła Informatyki
Gogołek Włodzimierz	Wydział Dziennikarstwa i Nauk Politycznych, Uniwersytet Warszawski
Hodoň Michal	University of Žilina, Slovakia
Jung Leszek	Instytut Sztuki i Nauk Technicznych, Społeczna Akademia Nauk
Kiedrowicz Maciej	Wydział Cybernetyki, Wojskowa Akademia Techniczna
Kołodziński Edward	Instytut Optoelektroniki, Wojskowa Akademia Techniczna
Komorowski Jacek	British Council, Warszawa

Korbel Piotr	Instytut Elektroniki, Politechnika Łódzka
Kowalski Andrzej	Instytut Telekomunikacji, Politechnika Warszawska
Kwiatkowski Włodzimierz	Wydział Cybernetyki, Wojskowa Akademia Techniczna
Liderman Krzysztof	Wydział Cybernetyki, Wojskowa Akademia Techniczna
Macukow Bohdan	Wydział Matematyki i Nauk Informacyjnych, Politechnika Warszawska
Mąka Wojciech	Wydział Nauk Technicznych, Uniwersytet Warmińsko-Mazurski
Murawski Krzysztof	Wydział Cybernetyki, Wojskowa Akademia Techniczna
Napieralski Piotr	Wydział Fizyki Technicznej, Informatyki i Matematyki Stosowanej, Politechnika Łódzka
Olchowik Wiktor	Wydział Informatyki, Wyższa Szkoła Technologii Informatycznych
Pałka Norbert	Instytut Optoelektroniki, Wojskowa Akademia Techniczna
Pałys Tomasz	Wydział Cybernetyki, Wojskowa Akademia Techniczna
Przelaskowski Artur	Wydział Matematyki i Nauk Informacyjnych, Politechnika Warszawska
Putz-Leszczyńska Joanna	Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska
Sienkiewicz Piotr	Wydział Bezpieczeństwa Narodowego, Akademia Obrony Narodowej
Sobieski Ścibór	Wydział Fizyki i Informatyki Stosowanej, Uniwersytet Łódzki
Sondej Tadeusz	Wydział Elektroniki, Wojskowa Akademia Techniczna
Stasiak Andrzej	Wydział Cybernetyki, Wojskowa Akademia Techniczna

Suski Zbigniew	Wydział Cybernetyki, Wojskowa Akademia Techniczna
Szafrański Bolesław	Wydział Cybernetyki, Wojskowa Akademia Techniczna
Ślusarczyk Grażyna	Zakład Projektowania i Grafiki Komputerowej, Uniwersytet Jagielloński
Świerczyński Zbigniew	Wydział Cybernetyki, Wojskowa Akademia Techniczna
Welker Elżbieta	Wydział Nawigacji i Uzbrojenia Okrętowego, Akademia Marynarki Wojennej
Weydman Romuald	MILSTAR
Wiśniewski Andrzej	Wydział Cybernetyki, Wojskowa Akademia Techniczna
Wojciechowski Adam	Wydział Fizyki Technicznej, Informatyki i Matematyki Stosowanej, Politechnika Łódzka
Żorski Witold	Wydział Cybernetyki, Wojskowa Akademia Techniczna
Żurek Leopold	MILSTAR
Życzkowski Marek	Instytut Optoelektroniki, Wojskowa Akademia Techniczna