SCUID^{Sim}: A Platform for Smart Card User Interface Research, Development and Testing

Markus Ullmann

Federal Office for Information Security
D-53133 Bonn, Germany
www.bsi.bund.de
and
University of Applied Sciences Bonn-Rhine-Sieg
www.h-brs.de
Email: markus.ullmann@bsi.bund.de

Ralph Breithaupt

Federal Office for Information Security
D-53133 Bonn, Germany
www.bsi.bund.de
Email: ralph.breithaupt@bsi.bund.de

***Abstract*—The latest advances in the field of smart card technologies allow modern cards to be more than just simple security tokens. Recent developments facilitate the use of interactive components like buttons, displays or even touch-sensors within the cards body thus conquering whole new areas of application. With interactive functionalities the usability aspect becomes the most important one for designing secure and popularly accepted products. Unfortunately the usability can only be tested fully with completely integrated hence expensive smart card prototypes. This restricts application specific research, case studies of new smart card user interfaces, concerning applications and the performance of useability tests in smart card development. Rapid development and simulation of smart card interfaces and applications can help to avoid this restriction. This paper presents SCUID^{Sim} a tool for rapid user-centric development of new smart card interfaces and applications based on common smartphone technology.**

Keywords–Smart Card; User Interface Design, Interactive Smart Card Applications; Rapid Prototyping.

I. INTRODUCTION

Recently developed interactive components allow the integration of input devices, like buttons, keypads or gesture interfaces as well as output devices like displays and LEDs directly into a smart card. This offers especially new security services like "on-card" user authentication and trusted displays and avoids the use of external terminals which are potentially vulnerable to active and passive attacks.

With the interactive functionalities the usability aspect becomes the most important one for designing a usable smart card and adds many new demands to the development process. Now aspects like the adequate size of a button, the visibility of a touch interface, the resolution, contrast and speed of a display and the overall design of the card have to be addressed as well as an appropriate hardware/software-codesign to ensure clear user guidance and high overall usability. This can only be achieved by conducting extensive field tests with as many different people as possible. Creating the necessary card prototypes with the complete design and full hardware and software functionality can be very expensive and time consuming which makes usability centered security research difficult. In this paper we present an alternative approach to allow the necessary testing in order to determine the requirements for design, hardware components and the software without the need to build costly prototypes. By using common smartphones almost all user related aspects can be investigated by simulating the "look & feel" of a new smart card design before any real hardware integration is needed.

SCUID^{Sim} is an android application and therefore usable on a wide range of smartphones which combine in a single compact device all the necessary hardware input/output components as well as communication links, cryptographic services, the processor power and memory needed for simulating a large variety of current and future smart card interfaces and applications. With SCUID^{Sim} the visible aspects of a multi-component smart card can be designed on the smartphone. Based on a simple SCUID^{Sim}-API, user defined card applications can be executed while SCUID^{Sim} simulates the behavioural properties of all interactive components. New requests and requirements can be implemented, simulated and evaluated instantly. This way SCUID^{Sim} supports detailed requirement engineering for software as well as hardware and the development of new user interface concepts hand in hand. This is especially useful for the design and integration of new usable user centric security algorithms in smart cards.

The rest of the paper is organized as follows: Section II starts with a description of related work. Section III provides a brief overview of the software architecture of SCUID^{Sim} and its functionality. Next, Section IV describes a first case study of a contactless smart card with a low cost user interface. The user interface consists of a touch slider component for user input and a display component implemented as 3×5 LED matrix which can only illustrate one character with very limited details. The use case "on-card" user authentication shows how concepts for user guidance and visual user feedback can be tested and evaluated in SCUID^{Sim}. Finally, Section V summarizes the findings of this paper and gives an outline of open issues regarding SCUID^{Sim}.

II. RELATED WORK

The first research and development projects investigating the idea to integrate input and output elements in smart cards go back as far as the late 1990s, see [1]. With the advances in low power and low profile embedded technologies

 ISBN: 978-1-61208-356-8

many different component technologies have been successfully developed and integrated in ID1-compatible smart cards during the last decate. Primarily a variety of display types and buttons even fingerprint scanners are discussed for integration, see [2] and [3]. Moreover, in [4] smart cards with an integrated display as security enforcing component are introduced. A first approach to integrate a 2D on-card gesture input sensor implemented as capacitive touch matrix is introduced first in [5]. It has also been an important topic for public funding in many countries (e.g. the INSITO-project of the german federal office for information security and the SECUDIS-project of the german ministry of education and research, see [6], and [7]). Despite all the effort and the growing number of available components, interactive smart cards have not yet been used in many real applications. Amongst other reasons this is due to high production costs and the much higher complexity of such smart cards. With the recent advances in printed electronics capacitive sensors have become standard technology and even printed displays are available today, see [8], [9], and [10]. But the complexity issue is still a serious obstacle on the way to the final product. At least regarding the system integration issues of combining several hardware components there have been approaches for rapid prototyping tools. One of the first was the FlexCOS system suggested by Beilke at al. [11], which uses FPGAs for a very flexible and rearrangable interface to connect separate component prototypes into one complete system. Although this approach became standard procedure for many manufacturers and researchers it only covers the technological aspects. Such functional prototypes are much too bulky and fragile to conduct real world tests with many people in real application scenarios outside the lab. The usability aspects that first and foremost define how the smart card should interact and therefore what the requirements for the hardware and software components really are can not be tested without fully integrated and designed card prototypes. Unfortunately each version of real prototypes to test for user acceptance require huge expenses of time and money. This lack of end-user centered rapid prototyping tools was the starting point for the development of the SCUID^{Sim} tool. Simulation of user interfaces was very popular in the beginning of ubiquitous computing. One approach was the iStuff toolkit to support the development of user interfaces for the post-desktop age for multiple displays, multiple input devices, multiple systems, multiple applications, and multiple concurrent users, see [12]. Alternative technologies were developed by the Stanford Interactive Workspaces project for multi-person and multi-device collaborative work settings, see [13]. To the best of our knowlege SCUID^{Sim} is the first approach to model, simulate and analysis user interfaces for (contactless) smart cards.

III. SCUID^{Sim} Architecture

SCUID^{Sim} consists of two modules: a card designer which enables a flexible but simple arrangement of smart card layouts based on preconfigured components and a card simulator. In the card simulator such a card layout can be paired with a smart card application in a real time simulation. It was a design decision to separate the card design process and the card simulation process in two independant software modules. Figure 1 illustrates the SCUID^{Sim} architecture.

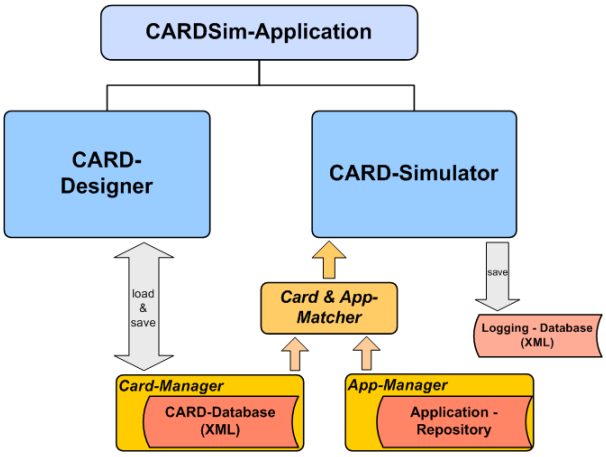

Figure 1. Overview of the SCUID^{Sim} software architecture.

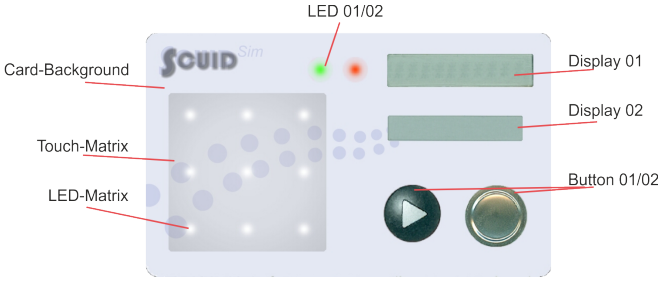

Figure 2. Available card components (in the card designer)

A. Card Designer

The card designer is a simple tool to engineer smart card layouts. Figure 2 gives an overview of the available components in the current version of SCUID^{Sim}. Currently the following predefined components: push buttons, segmented displays (7- and 14-segments), matrix displays (RGB, greyscale and black & white), LEDs, n × m LED-matrixes, 2D-touch sensors, image boxes and the overlay image of the smart card are supported. There are also non-visible components like e.g. acceleration sensors that are automatically available to all cards if the used android smartphone is supporting it. So based on this predefined components, SCUID^{Sim} can simulate a huge variety of smart card layouts. Figure 3 depicts a real card prototype opposite to a replicated design of this card within SCUID^{Sim}. This Figure illustrates the very realistic replication capabilities of our tool.

Within the card designer the properties of each compo-

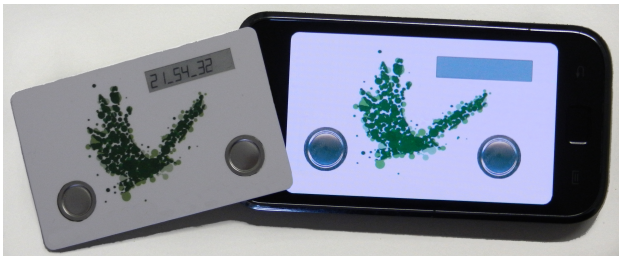

Figure 3. Confrontation real - and simulated card layout within SCUID^{Sim}

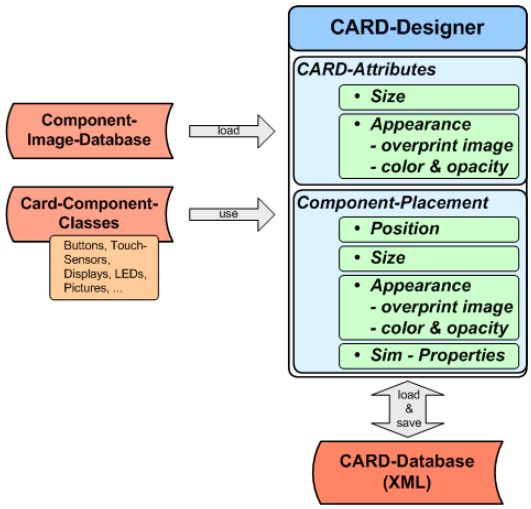

Figure 4. Software architecture of the card designer module

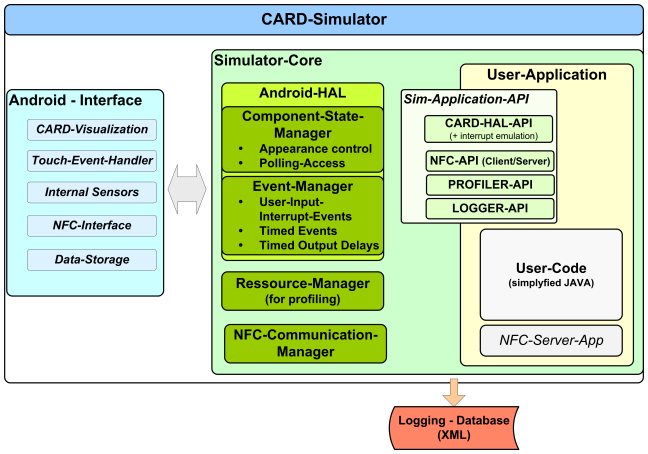

Figure 5. Software architecture of the card simulator module

nent like selection, position, size, and deletion can easily be controlled via simple finger gestures commonly known from many other mobile applications. Additional properties like overlay image (appearance of the component), color modifier (to the overlay image, in RGB and alpha for transparency) or component specific properties like X/Y-resolution of a matrix display, or update delay time for a display component can be set in a component property page that is dynamically generated based on all the properties of a selected card component. Each card component, its properties and its specific simulated behaviour (e.g. delay of the visual update) is defined in the respective class within the component library of SCUID^{Sim}. To add new components or behavioural functionality to this library the developer simply inherits and modifies the provided component base class. All administrative support like the list of available component types and the components property page are generated "on the fly". The complete card design can be loaded from and saved to a card library in an XML-format that can be read and edited outside SCUID^{Sim} with all existing standard XML-viewers/editors. Figure 4 depicts the software architecture of the card designer.

B. Card Simulator

The two main objectives of the card simulator are to provide a flexible framework for the development and evaluation of card applications and to simulate an user interaction as close to a real smart card as possible. For creating card applications the card simulator offers a simple API in order to access the interactive components of the simulated card. In order to keep as close to a real card program as possible the API allows input components to be polled as well as providing a simulated interrupt event handling. The concept of the API is based on the intention to shield the application developer from Android Java specific constructs in order to facilitate application code that can easily be transferred to real smart cards. In addition the card simulator consists of a ressource manager module for simple profiling purposes as well as a flexible logging system. Since most applications for contactless smart cards implicate a communication to a reader/server via NFC (ISO 14443) the card simulator offers an interface to the real NFC component of a smartphone. This way the simulated card can also be used in the targeted environment. If the real NFC component cannot be used, the framework allows the execution of NFC server applications in order to also simulate the reader functionality. A manual describing the usage and programming of applications can be found in [14]. Figure 5 depicts the software architecture of the card simulator module.

IV. Case Study: Smart Card with 3 × 5 LED Matrix Display

In this Section a contactless smart card with a very restricted user interface is described and analyzed. Typically contactless cards follow the ISO 14443 specification, see [15]. This means that contactless smart cards have no battery. They are powered by a magnetic field of a terminal device. So the available energy on real contactless smart cards for powering additional components is very limited. Due to the very limited available power the user interface consists only of a 3 × 5 LED matrix as an information display and an additional touch slider component for controlling user inputs. Wiping enables scrolling the characters of the alphabet and a long touch ($\geq$ 2 second) selects the denoted character in the display. First user tests have shown that a matrix display with less then 3 × 5 LEDs reduze the readability of characters seriously. So from a readability perspective a 3 × 5 LED matrix display seems to be a minimum requirement. To power a real contactless card with a 3 × 5 LED matrix display and an additional touch component is technically possible but from an energy perspective still a challenge. Figure 6 describes the card layout of the analyzed card.

Here we analyse the described user interface especially for performing an user authentication process on-card. The user authentication is performed based on a shared secret (classical password which consists of a sequence of 4 or 6 digits, e.g. *5839*). The reference password is already configured and securely stored in the card memory. Here only the user authentication process itself without additional services, like changing the user passwords, etc. is regarded. Further security aspects of smart cards and smart card application like secure storage of the password and other secrets, used cryptographic protocols or side channel free implementation of cryptographic algorithms are not subject of this paper. Here we refer to [16] or [17]. The main issue of the use case is the demonstration

 ISBN: 978-1-61208-356-8

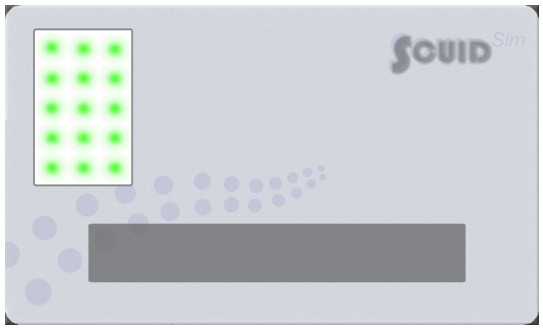

Figure 6. Layout of the smart card used in the case study (3 × 5 LED matrix display and slider component)

of the capabilities of SCUID^{Sim} to analysis user interfaces.

Concerning the user interface following questions arise:

- Which alphabet is useable with a 3 × 5 LED matrix display ?
- How can status information of the application and necessary feedbacks be given to the user, e.g. feedback of a character selection or a successful respective failed password authentication ?

A. On-card User Authentication Process

The authentication process has three security states: *in authentication — authenticated — locked*. If the card is wattless or the authentication process is still running the card is in the security state *in authentication*. If the user authentication is performed successfully the card switches to state *authenticated* and further card applications (which are not described here) can be performed. After card processing the security state switches to *in authentication* again. If the user exceeds the maximum number of authentication retries during the authentication process the card is blocked and switches to security state *locked*. In Figure 7 the whole authentication process is depicted. The security states are highlighted as trapeziums and have to be visualized to the user by an adaquate shown symbol or text.

The user authentication process starts with displaying the security state *in authentication*.

Next, one character of the used alphabet is displayed. Each selected character of the user (long user touch ≥ 2 seconds) is shortly displayed (as user feedback) followed by illustrating the number of already inserted password characters depicted as understandable symbols. If the live password has completely been entered (e.g. 4 digits) the password verification (live password $\stackrel{?}{=}$ reference password) is performed automatically. A successful user authentication has to be depicted clearly to the user. If the password authentication fails the second authentication attempt starts automatically. This has to be shown to the user. If the second authentication attempt fails again the last attempt (assumption: three password attempts) is performed which has to be denoted to the user as well. If the final authentication fails again the card switches to the security state *locked* and no user operation is possible any more. This security state has also to be displayed to the user. The white boxes in the Figure 7 represent position of the authentication process where feedback information have to be given to the user by text outputs or specific symbols.

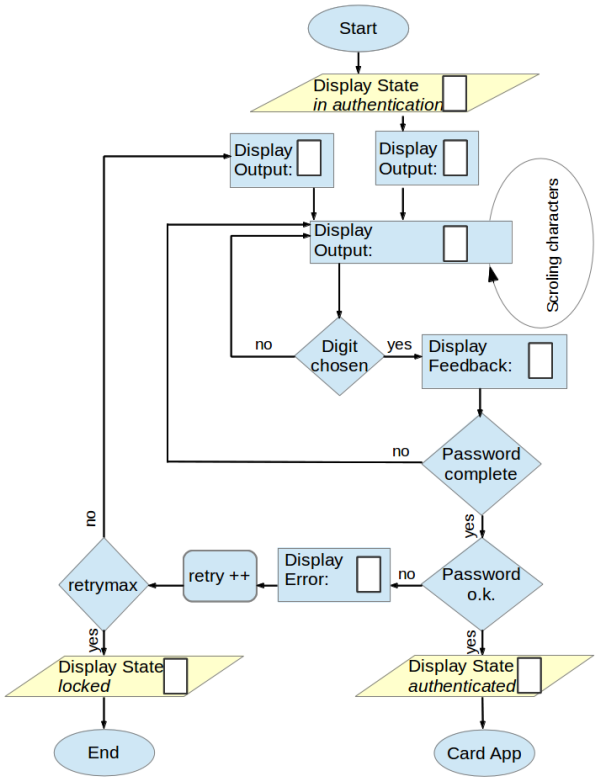

Figure 7. Authentication process

Summing up, following situations and security states have to be shown to the user:

- current security state *in authentication — authenticated — locked*
- number of already entered characters
- successful resp. failed user authentication
- current authentication round (status password retry counter)

In addition following information like: application start, application end, card in processing, etc. seems to be important information for the user and has to be displayed, too.

B. General Display Illustration Facilities

First, the principle illustration facilities of a 3 × 5 LED matrix display are presented. Static characters:

1) Characters, e.g. alphabet shown in Figure 8
2) Special characters, e.g. dice symbols shown in Figure 9
3) Symbols, e.g. arrows, rectangle, box, horizontal and vertical lines, ...

Animated symbols:

4) Special characters, e.g. falling dice symbols
5) Symbols, like falling arrow (picture frequency 200 ms), curtain up (picture frequency 200 ms), curtain

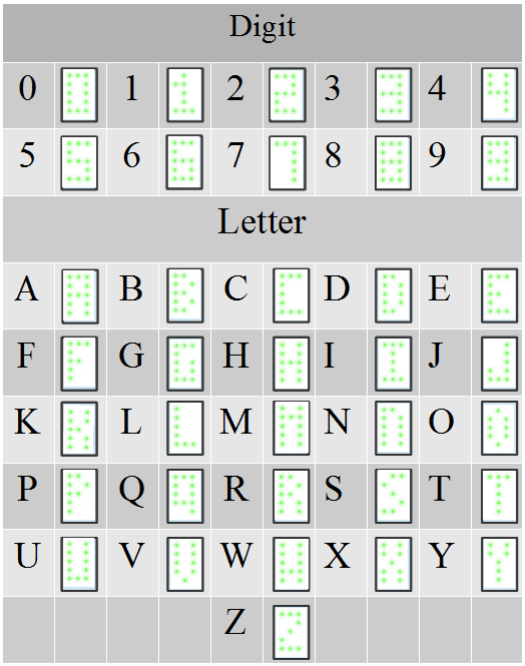

Figure 8. Alphabet

Figure 9. Dice symbols

down (picture frequency 200 ms), and rotary dots (dot frequency 200 ms) shown in the first row from left to right in Figure 10 and helix construction (sequentially build up dot by dot with dot frequency 200 ms), helix destruction (sequentially build up dot by dot with dot frequency 200 ms), o.k. symbol (sequentially build up dot by dot with dot frequency 200 ms), and fail symbol (sequentially build up dot by dot with dot frequency 200 ms) shown in the second row from left to right in Figure 10.

This listing is not complete. But it shows that even the very restricted display enables the presentation of a large range of characters and symbols especially when static and dynamic (animations) effects are exploited.

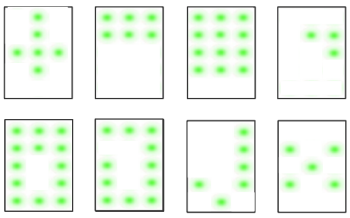

Figure 10. Animated symbols

C. User Test

First user test with only fourteen persons (students in age 20 - 30: 2 females, 12 males) have been performed. Obviously that is no adequate user cross section and no statistical relevant number of attendees. However first interesting user perceptions can be given.

The used digits of the alphabet shown in Figure 8 are distinguishable. But as seen in Figure 8 some characters are only poorly distinguishable in a 3×5 LED matrix display setting, e.g. B, G, N, O, P, Q, X. Lower letters worsen the problem dramatically. This means if only digits are processed a 3×5 LED matrix seems to be sufficient. But if letters should be processed too higher resolution displays e.g. a 5×5 LED matrix display or a 4×7 LED matrix display is needed to achieve better letter readability.

Additionally we tried to output short words (e.g. on, off, o.k., ...) as feedback to the user by sequentially diplaying the characters of the word. The users have enormous problems to read and identify even very short words depicted as sequence of letters when they do not know the displayed word before. The consequence is that this approach for displaying words can not be followed anyway in a 3×5 LED matrix display setting.

On the contrary animated symbols like, falling arrows and rectangles, dynamic curtain, circling dots etc. seem to be very intelligable to the user. Animated symbols seem to be a suitable alternative to text output to indicate card states and to give feedback information to the user. So we performed additional user tests concerning animated symbols.

We used our sample card (see Figure 6) to show the participants sequentially animations of the symbols displayed in Figure 10 in an unsorted order to link the symbols to predefined meanings. We applied following procedure: Each symbol is animated first and afterwards shown for 5 seconds to the participant as a static symbol before the animation of the next symbol starts. This procedure was performed for 5 minutes. During this test the attendees should match an animated symbol to one of the given meanings: authenticated — locked — in authentication — start application — end application — card in processing — password check o.k. — password check fails — first authentication attempt. The result was very heterogeneous. Apart from password check o.k. (o.k. symbol) and password check fails (fail symbol) with nearly 40 % correct assignments there was no significant occurency of any symbol meaning. The mappings (meaning $\leftrightarrow$ symbol) of the participants were very scattered. The finding is that even symbols need to be chosen very carefully and have to be explained to the user in detail. If it is possible to use symbols which are intelligible to all they should be applied in any case.

The consequences for our application with a 3×5 LED matrix setting are therefore:

- only digits should be used as alphabet
- text outputs as user guidance is not possible instead (animated) symbols should be used
- we prefer dice symbols to denote the number of already inserted characters of the password and
- animated symbols for indicating the security state and for arbitrary feedback to the user.

 ISBN: 978-1-61208-356-8

But which animated symbol should be used for indicating the current security state *in authentication — authenticated — locked*, already performed authentication attempts . . . is still an open issue and is subject of further examinations.

V. CONCLUSION

In this paper we present our smart card user interface development and simulation tool SCUID^{Sim}. This tool enables the rapid development and simulation of smart card interfaces and applications. It can be used for user interface research, easy prototyping and performing of tests.

Therefore this tools is useable for early consideration of user handling requirements and overall user acceptance of user interfaces before a time consuming and costly prototype development starts. Especially, card designs and application modifications are performed very quickly in software without any hardware modification. This reduces the development of smart card prototypes and speeds up the whole development process. Moreover the tool is very useful for the design and exploration of new usable security concepts and algorithms for contactless cards and enables further application specific research in this direction.

SCUID^{Sim} is available in version 0.3. After further useability studies it is planned to implement additional functionalities. At the moment microphones, cameras, initial sensoric and biometrics are not supported. Here, new card component classes have to be developed and integrated in SCUID^{Sim}.

Furthermore, SCUID^{Sim} is a whole framework. Card user studies do not require the whole functionality of the framework. A specific SCUID^{Sim}-subset is sufficient. This subset of SCUID^{Sim} is currently not available, too.

The test of new interface concepts requires the consideration of different user groups (range of user ages, sex, . . .). To achieve a good test performance and test analysis a central logging of the test data and user feedbacks is desirable. Furthermore an automatic code update for all smartphones involved in an evaluation is preferable. That is subject of a further enhancement, too.

Within this paper a case study for a very restricted user interface for smart cards is described together with first user test results. This case study depiced in Section IV presents the potential of SCUID^{Sim} to analyse new user interface approaches. The analyzed user interface consists of a 3×5 LED matrix display and a slider component. This setting was used due to energy restrictions for additional components in real contactless cards. The test results show that a 3×5 LED matrix display is not adequate do display letters of an alphabet. It is sufficient to display digits. But it show that even the very restricted display enables the presentation of a large range of symbols when static and dynamic effects are exploited. This first results are used to guide further studies of smart user interfaces for contactless smart cards.

VI. ACKNOWLEDGEMENT

The authors would like to thank our students Anton Buzik, Alexander Kreth and David Sosnitza for their support implementing SCUID^{Sim}, and our colleague Christian Wieschebrink for valuable remarks. Also thanks to the anonymous reviewers for the valuable comments.

REFERENCES

[1] Gerald V. Piasenka, and Thomas M. Fox, and Kenneth H. Schmidt, "Solar cell powered smart card with integrated display and interface keypad," 1998, US patent US5777903.

[2] J. Fischer, F. Fritze, M. Tietke and M. Paeschke, "Prospects and Callenges for ID Documents with Integrated Display," in Proceedings of Printed Electronics Europe Conference, 2009.

[3] Bundesdruckerei, "RFID Security Card with a One-Time Password and LED Display," 2013, www.rfidjournal.com/articles/10512.

[4] M. Ullmann, "Flexible visual display unit as security enforcing component for contactless smart card systems," in Firth International EURASIP Workshop on RFID Technology (RFID 2007), 2007, pp. 87–90.

[5] M. Ullmann, R. Breithaupt, and F. Gehring, "On-card user authentication for contactless smart cards based on gesture recognition," in Proceedings GI Sicherheit 2012, ser. Lecture Notes In Informatiks, no. 108, 2012, pp. 223–234.

[6] BDR, "Secudis Project," 2012, http://www.bundesdruckerei.de/en/684-innovative-high-security-solutions.

[7] "Thin chips for document security," in Ultra-thin Chip Technology and Applications, J. Burghartz, Ed., 2011.

[8] Taybet Bilkay, and Kerstin Schulze, and Tatjana Egorov-Brening, and Andreas Bohn, and Silvia Janietz, "Copolythiophenes with Hydrophilic and Hydrophobic Side Chains: Synthesis, Characterization, and Performance in Organic Field Effect Transistors," Macromolecular Chemistry and Physics, vol. 213, September, 26 2012, pp. pp. 1970–1978.

[9] P. Andersson, R. Forchheimer, P. Tehrani, and M. Berggren, "Printable all-organic electrochromic active-matrix displays," Advanced Functional Materials, vol. 17, no. 16, 2007, pp. pp. 3074–3082. [Online]. Available: http://dx.doi.org/10.1002/adfm.200601241

[10] PaperDisplay, "Printed Display Products," 2013, http://www.paperdisplay.se.

[11] K. Beilke and V. Roth, "Flexcos: An open smartcard platform for research and education," in Proceedings of the 6th International Conference on Network and System Security, NSS 2012, ser. Lecture Notes In Computer Science, no. 7645, 2012, pp. pp. 277–290.

[12] R. Ballagas, M. Ringel, M. Stone, and J. Borchers, "istuff: a physical user interface toolkit for ubiquitous computing environments," in Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 2003, pp. 537–544.

[13] J. Borchers, M. Ringel, J. Tyler, and A. Fox, "Stanford interactive workspaces: a framework for physical and graphical user interface prototyping," Wireless Communications, IEEE, vol. 9, no. 6, 2002, pp. 64–69.

[14] BSI, "$SCUID^{Sim}$ manual, version 0.3," 2013.

[15] ISO/IEC, "ISO/IEC 144443 contactless Integrated Circuits Cards, Part 1-4: Physical Characteristics (1), Radio Frequency Power and Signal Interface (2), Initialization and Anticollision (3) and, Transmission Protocol (4)," 2000.

[16] W. Rankl and W. Effing, Smart card handbook. John Wiley & Sons, 2010.

[17] S. Mangard, E. Oswald, and T. Popp, Power analysis attacks: Revealing the secrets of smart cards. Springer, 2008, vol. 31.

 ISBN: 978-1-61208-356-8