Systems Security Requirements Analysis for M-Government Transformation

Fatima Al Shamsi, Sarah Bamatraf, Talal Rahwan, Zeyar Aung, and Davor Svetinovic
Department of Computer Science, Masdar Institute
Khalifa University of Science and Technology, Abu Dhabi, UAE
{fsalshamsi, sobamatraf, trahwan, zaung, dsvetinovic}@masdar.ac.ae

***Abstract*—With the launch of smart government initiatives, it is important to analyze the security requirements that will enable the decision makers to successfully carry out the process of transformation to mobile government (M-Government). This paper presents a security analysis using the security requirements engineering SQUARE method in the context of the M-Government transformation. The mobile application architectures are evaluated as a case study using the SQUARE method. In particular, the main contribution of this paper is the analysis of the outcomes of the SQUARE method using qualitative evaluation criteria.**

***Index Terms*—security requirements engineering, smart government,**

I. Introduction

With the ever-increasing popularity of smart mobile devices, mobile government (M-Government) initiatives continue to be launched in various countries around the world. For example, in the United Arab Emirate (UAE), many initiatives of smart government such as, e.g., mobile government (M-Government) and electronic government (E-Government) have been launched to advance the life of UAE residents. The aim is to realize a government that is accessible to all individuals in UAE, a government that can be available at all times, and a government whose services can be accessed via smart devices [1].

A successful deployment of the initiative could potentially facilitate people's lives, and would ultimately lead to a government that maintains a global presence of the UAE in the technological fields, while practicing and complying with the international standards. More specifically, the UAE government has set a roadmap with four parallel tracks [1]:

1) establish an environment for M-Government;
2) assess the capability and capacity of government entities;
3) establish shared resources across government entities at the national level;
4) achieve citizens' happiness.

The complexity, technical difficulty and shared services among government entities pose great risks that could hamper the process of transforming to M-Government. In particular, an erroneous or substandard implementation could significantly hold back the entire process of transformation. Even some of the M-Government system's components, which are initially thought to be irrelevant to security, may eventually turn out to compromise the security when combined with other components. This is especially true since, in an interconnected system, the requirements and components are likely to be identified by multiple stake holders with varying, and perhaps even conflictive, interests. In order to prevent any future security vulnerabilities, any potential security issues should be detected early, during the system-development process [2].

The Security Quality Requirements Engineering (SQUARE) method [3] integrates security requirements engineering (RE) into the software development process. It provides a conceptual and organizational framework for implementing RE with an emphasis on security [4]. SQUARE helps in categorizing and prioritizing security requirements for ICT systems and applications. It also facilitates the integration of security considerations in the early stages of the system development life cycle. Furthermore, SQUARE can be used for documenting and analyzing the security requirements for existing systems, which can lead to the improvement and modification of these systems.

The research objective of this paper is to assess the effectiveness of SQUARE, and to apply it to the M-Government transformation as a case study. Using SQUARE, we elicit the possible set of threats, vulnerabilities, and risk analysis of M-Government transformation, and then propose a categorization and prioritization of the security requirements for such transformation. The assessment of SQUARE is carried out using 12 criteria for evaluating a security RE process.

II. Related Work

RE is considered a critical component in the process of system development. There is a pressing need for developing a model that has the ability to scrutinize the security and quality requirements during various stages of the system's development life cycle. One important security RE method is the Security Requirements Engineering Process (SREP) — an asset-based and risk-driven method that is used for the purpose of establishing security requirements during the process of developing secure information systems [5].

Mellado, Fernández-Medina, and Piattini [5] emphasized the significance of identifying security quality requirements when developing a security critical information system. The authors presented a common criteria-centered and reuse-based process that addresses the security requirements at the early stages of system development in an organized and intuitive manner. This way, the security resources repository and the

978-1-5386-3337-3/17/$31.00 ©2017 IEEE

common criteria are integrated into the system development life cycle thus unifying the processes of RE and security engineering.

Biel, Grill and Gruhn [6] presented a methodology that addresses a software systems' usability holistically by aligning software architecture (SA) exploration and usability evaluation (UE). The authors exemplified how these techniques can be shared and how their results correlate with each other. They proposed a methodology that regards requirements as a common basis for both SA analysis (an inspection method) and UE analysis (a user test) and discusses the results in the form of usability issues. As a case study, they categorized and defined elements that describe usability in the context of a mobile application system.

Van Lamsweerde [7] developed a constructive approach that models specifications and analyzes application-specific security requirements. It is based on a goal-oriented framework that generates and resolves obstacles to reach the goal-satisfaction. This study addressed malicious obstacles called *anti-goals* which are set up by attackers to threaten the system's security goals. Threat trees are constructed through anti-goal refinement. This process produces leaf nodes that are either software vulnerabilities that the attacker can recognize, or anti-requirements that the attacker can implement. New security requirements are then identified as countermeasures by using threat resolution operators derived from the anti-requirement and vulnerability specifications of the analysis.

Mead and Stehney [3] focused on eliciting and prioritizing security requirements of system development within organizations. Two case studies were investigated, where the model was applied to the client system. The paper also proposed the *Security quality requirements engineering methodology (SQUARE)* [3], which incorporates security RE into the software development process. It provides a conceptual and procedural framework for implementing security RE activities. We use SQUARE for our M-Government transformation case study in this paper.

III. Research Method

The main contribution of this work is to investigate M-Government transformation from security viewpoint, whereby security risks and requirements analysis for a successful M-Government transformation are studied. Our case study encompasses a comprehensive list of tasks required to complete the research. These tasks include designing the case study, collecting the data, analyzing the data, and presenting and evaluating the results.

Our research objective is to assess the effectiveness of the SQUARE method [3] and its application on M-Government transformation as a case study. This is done through the scrutiny of well-defined sets of threats, vulnerabilities, risk analysis, and categorization and prioritization of security requirements. SQUARE is composed of nine steps: (1) agree on definitions, (2) identify security goals, (3) develop artifacts to support security requirements definition, (4) perform risk assessment, (5) select elicitation techniques, (6) elicit security requirements, (7) categorize requirements, (8) prioritize requirements, and (9) requirements inspection [3].

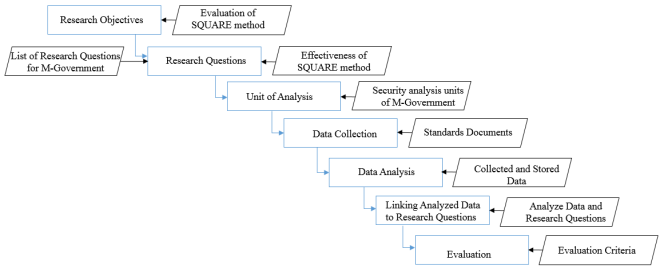

Figure 1. Case Study Design Plan

Our assessment of SQUARE is carried out using 12 security RE evaluation criteria, namely, (1) flexibility, (2) sampling, (3) analyst permissibility, (4) interpretiveness, (5) data sufficiency, (6) coherence, (7) repeatability, (8) usability, (9) adaptability, (10) complexity, (11) implementation duration, and (12) flexibility.

The data used in the case study are acquired from reliable sources, namely academic and industrial publications and documents. Then, the data is analyzed and presented in a systematic manner. Our research focuses on an explanatory sequential mixed method where quantitative research is conducted and analyzed first, before conducting the qualitative research. This approach is considered explanatory as the initial quantitative data results are explained further with the qualitative data [8]. The design plan of the case-study research carried out in our study is illustrated in Figure 1.

IV. Results

In this section, we present the results of the nine SQUARE steps on the M-Government transformation case study.

Step 1 - Agree on Definitions: We first defined a number of concepts related to security, namely: (1) risk analysis, (2) authentication, (3) authorization, (4) availability, (5) data security, (6) security level, (7) transmission security, (8) vulnerability assessment, and (9) contextual integrity. We generally adopt the definitions given in [9]; these will not be reproduced here due to space constraints.

Step 2 - Identify Security Goals: Three main security goals were identified to serve the main business goal of M-Government. In particular:

- *Business goal (B-01)*: The mobile application allows the users (i.e., clients) to make informed decisions based on the information and other assets available and ultimately achieve the following security goals.
- *Security goal (G-01)*: The system's administrator must be able to exercise effective control over the system's configuration and usage.
- *Security goal (G-02)*: The system's confidentiality, integrity, and data accuracy must be preserved.
- *Security goal (G-03)*: The system must be available to the user whenever needed.

Table I
USE CASE 1: PAYING TRAFFIC FINES M-PAY

UC-01	Paying traffic fines using transactional mobile app
Description	All users who can access the application will be able to pay for traffic fines using their credit/debit card.
Delivery context	G2C
Actors	User and System Administrator
Assumptions	1. System is available 2. Mobile Application is connected to the internet. 3. Transaction is completed via a secure network (SSL). 4. User correctly entered username and password.
Steps	1. User registers his credentials in the application or the website by providing personal details [10]. 2. User registers payment details in the application. 3. User completes the activation process. 4. User enters his/her username and password to access mobile application services. 5. User manages subscription of service offered by mobile application. 6. User views/selects traffic fine that he/she wants to pay. 7. User clicks pay to complete the service.
Consequence Variations	Once logged in, the user is able to click on *My Account* tab and choose the auto payment option [10].
Non-Functional	They do not have any edit privileges They will be assigned view-only privileges. Transaction is only completed if user is authenticated [11]. If the user fails to enter correct payment details, the application will display a pop-up message stating that the provided details are not valid.

Step 3 - Develop Artifacts: We specified three artifacts: (1) system architecture diagram, (2) use cases, and (3) use case diagrams. The mobile security architecture in Figure 2 provides an overview of the essential components needed to provide a secure mobile service. Confidentiality, integrity, and availability are critical for the success of government service. The figure is an expansion on the notional mobile architecture, yet it covers a broad range of mobile applications and services which supports multiple use cases. It describes the components of the infrastructure of mobile computing, such as MDM, MAM, IAM, data management services, VPNs, firewalls, intrusion detection systems, and security gateways. These components mediate access from the mobile computing infrastructure to the enterprise information and service infrastructure.

To describe the sequence of actions that might arise while using M-Government applications, use cases were used for that purpose. The features of the developed use cases included: the use case description, delivery context, actors, and expectations of the existing state of the application prior and while being used by the actor. Other features included the steps taken by the actor in order to finish the service, as well as the consequences, which are separated into two main categories: variations and non-functional.

As a concrete example, we present a M-Government use case. Table I refers to the use case, named *M-Pay*, which is the transactional service of paying traffic fines where the delivery context is government to citizens.

Step 4 - Perform Risk Assessment: In order to determine the likelihood and the effect of M-Government application exposure, a risk assessment was performed. The Risk Management Guide for Information Technology Systems, NIST SP 800-30 [12], was employed. Step 4 is one of the most significant steps and it comprises of threat identification, vulnerability identification, control analysis, likelihood determination, impact analysis, and finally risk determination.

Table II
M-GOVERNMENT SECURITY THREATS

Threat Number	Threat Description
T-01	Insecure client-side data storage
T-02	Lack of data protection in transit
T-03	Personal data leakage
T-04	Resources with weak authentication
T-05	Failure to incorporate least privilege authorization policy
T-06	Client-side injection
T-07	Client-side Denial of Service Attack (DOS)
T-08	Malicious third-party code
T-09	Client-side buffer overflow
T-10	Failure to apply server-side controls
T-11	Failure of properly managing inbound SMS messages
T-12	Failure of properly managing outbound SMS messages
T-13	Ability of one application to view data or communicate with other applications
T-14	Data altered or observed during transaction in an unencrypted transactional channel
T-15	Failure to protect sensitive data at rest (in in databases or repositories), in use or in motion to unauthorized parties
T-16	Failure to disable insecure platform features in application (caching of keystrokes, screen data)
T-17	Improper categorization of sensitive data
T-18	Failure to provide usability
T-19	Exploitation of weaknesses in a database's development environment
T-20	Failure to monitor third party access to data repositories
T-21	Failure to log and monitor inappropriate sensitive data transfers
T-22	Failure to prevent users from uploading data to the web using online backup tools
T-23	Failure to implement encrypted and restricted remote access
T-24	Failure to apply applications control to restrict user capabilities such as copy and paste
T-25	Failure to restrict access to local admin functions (end point security)
T-26	Failure to encrypt hard disks at database servers
T-27	Failure to manage access to network-based repositories containing sensitive data
T-28	Failure to apply session handling features
T-29	Risk of side channel data leakage
T-30	Failure to apply sufficient transport layer protection
T-31	Failure to follow platform standard use published guidelines
T-32	Failure to log users out of mobile application
T-33	Failure to prevent code tempering / reversing
T-34	Transmitting data in plain text
T-35	Transmitting data using weak protocols
T-36	Transmitting data using weak ciphers/encryption protocols that are easy to break

The likelihood that a possible vulnerability could be implemented by a given threat source can be labeled as either high, medium, or low. *High* denotes that a threat source can be highly motivated and adequately able to control the weakness from being exercised. *Medium* states that the threat-source is driven and capable, nonetheless, it may impedes successful exercise of the vulnerability. *Low* indicates that the threat-source lacks the motivation and capability, and that controls may avert, or at least hinder, the vulnerability from being exploited [12].

Computing the level of risk is significant as it determines the adverse impact subsequent from a successful threat exercise of a certain vulnerability. The opposing effect of a security event is defined in terms of loss or degradation of any combinations of any of the security goals of confidentiality,

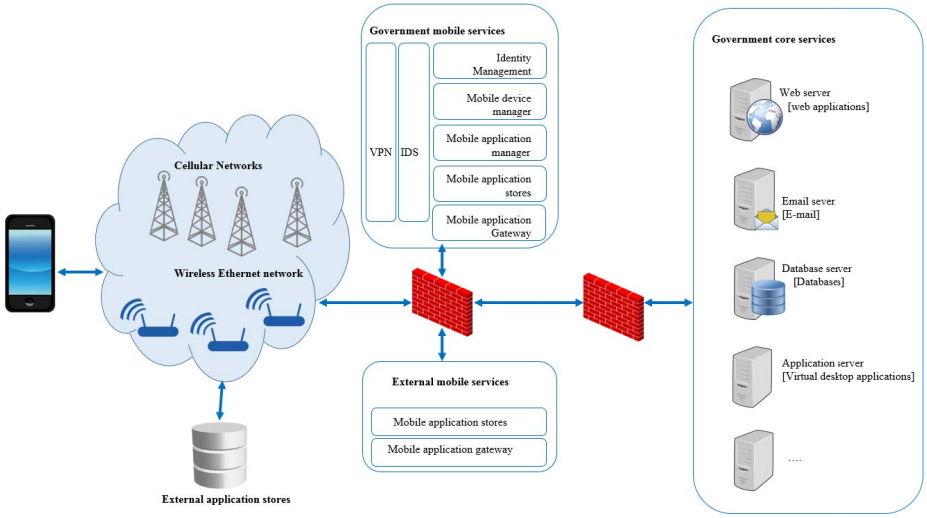

Figure 2. Mobile Security Reference Architecture

integrity, availability. The impact can also be lableled as either high, medium or low. In the high impact, the exercise of vulnerability might lead to a highly costly loss of major tangible assets or resources. In the medium impact, exercise or vulnerability may result in violating, or harming organizational mission, reputation, or interest. The low impact indicates that there is loss in some tangible assets or resources [12].

The ultimate fortitude of mission risk is derived by the process of multiplying the ratings assigned to threat likelihood and threat impact. In a 3×3 matrix containing threat likelihood (high, medium, low) and threat impact (high, medium, low), we assigned a probability that is related to each threat's likelihood level, with 1.0 for high, 0.5 for medium, and 0.1 for low. Also, we allocated for each impact level a value of 100 for high, 50 for medium, and 10 for low. The total risk scale is high if the overall value is from 50 to 100, medium if overall value is from 10 to 50, and low if overall value is from 1 to 10 [12].

If an observation assessed as high on the total risk scale, there is an urgent necessity for corrective measures. That is, the existing system might remain operating, but corrective actions must be carried out as immediately as possible. If an observation is assessed as medium on the total risk scale, corrective actions are required but not immediately necessary. As such, a plan must be developed to incorporate these corrective actions within a reasonable time frame. If an observation is assessed as low on the total risk scale, the system's administrator is obligated to determine whether corrective actions are required or whether it is possible to live with the risk [12].

We identified 36 possible threats towards mobile applications (Table II) and 36 possible vulnerabilities (omitted due to space limitations).

Steps 5 and 6 - Select Elicitation Techniques and Security Requirements: Given the M-Government values, business asset identification, and security objective determination, we applied the security requirements elicitation, thus obtaining a security requirements list (omitted due to space limitations). The list highlights 20 essential security requirements that we identified given the security threats and vulnerabilities.

Step 7 - Categorize Requirements: We categorized security requirements according to six categories: (1) confidentiality, (2) access control, (3) data integrity, (4) authorization, (5) usability, and (6) authentication.

Step 8 - Prioritize Requirements: the Analytic Hierarchy Process (AHP) methodology was applied to handle security requirements. The method calculated the relative values and the associated costs between security requirements. By applying the technique, we were able to confirm the consistency of our results, eliminate subjective judgmental errors, and increase the possibility that more reliable results are obtained. The value is measured on a scale from 1 to 9 [3].

After applying the AHP method on the security requirements, we found that the top 3 requirements are: R-09 with an estimated priority percentage of 15.9%, R-02 with an estimated priority percentage of 15.2%, and R-14 with an estimated priority percentage of 14.9%. In contrast, security requirements R-03 and R-01 scored the least priority, with an estimated priority percentage of 1.4% and 1.5%.

Given the result, confidentiality, integrity, and availability are of critical importance to the success of the M-Government transformation process.

Step 9 - Inspect Requirements: We inspected the elicited security requirements to make sure that they are correct, accurate and suitable.

V. Evaluation

In this section, we evaluate the results of applying the SQUARE method to the M-Government system transformation. In the analysis of M-Government applications, we discovered and specified use cases, security architecture diagrams, and use case diagrams. From the use cases, we elicited the

most critical security threats. Overall, we have discovered and specified 25 use cases, 36 threats (Table II), 36 vulnerabilities, and 20 main security requirements.

SQUARE is a method that is specifically designed for security RE. The method is composed of a variety of processes where misuse cases and use case diagrams are an essential part of the process. There are activities of high value such as identifying vulnerable, critical assets activity, and repository improvement activity which are missing from the SQUARE method. It is important to note that the elicitation approaches used in our paper do not result in a formal specifications. Resources used to gather data were from formal documentation, including IBM, Microsoft, and NIST 800-53 [13].

In order to evaluate the results of applying SQUARE on M-Government transformation, we assess the method using the following 12 criteria (only 6 presented due to the space limitations).

(1) Flexibility is evaluated by verifying the extent of which the SQUARE method enables us to identify the primary security goals, threats, vulnerabilities, and requirements. We were able to apply SQUARE on our case study given the flexibility of choosing from multiple techniques at each of the nine steps. Step one of SQUARE can be completed by techniques such as structured interviews or focus group between stakeholders and requirements team. In our case, we agreed on definitions by a focus group. Step two can be completed by applying techniques such as facilitated work sessions, surveys or interviews. In our case, we identified security goals by reviewing the business goal of M-Government and the types of services.

We were given the flexibility to use different input types in step three such as scenarios, misuse cases, templates and forms to produce artifacts that support the security requirements definition. We used use cases, use case diagrams, and security mobile application architecture.

We were required to complete step four by performing risk assessment. SQUARE provides a number of risk assessment techniques to choose from, each with a different level of difficulty. Risk assessment techniques that SQUARE team suggests are the GAO model [14], NIST model [12], SAEM [15], CMU's V-RATE method [16], Sage and Haimes' RFRM model [17], and Cornford et al.'s Feather's DDP model [18]. We selected the NIST model to be our risk assessment technique given the size of our case study. This model was found to be suitable for small scale projects. It was ranked highest given the criteria that it does not require additional data, and it is the most suitable for the requirements [3].

Steps five and six are about eliciting security requirements. They are usually completed by applying techniques such as Accelerated Requirements Method (ARM) [19], Joint Application Development (JAD) [20], surveys, interviews, checklists, model-based analysis, and document reviews. We completed step five of SQUARE by considering M-Government values and business goals and reviewing formal documents.

Steps seven and eight are about categorizing and prioritizing requirements. There are multiple techniques to prioritize requirements such as Triage [21], Win-Win [22], and the Analytic Hierarchy Process (AHP) methodology which we choose to apply in our case study [23].

Step nine is about requirements inspection, that is to review the requirements, and ensure that there are no defects such as ambiguities, inconsistencies, or mistaken assumptions.

Overall, we found that SQUARE offers flexibility in all of its steps, where the requirements team and stakeholders have the flexibility at each step of the process to select the most suitable method. It was also found that SQUARE has the flexibility to be implemented at any stage of the project development life cycle. In our case study, the government has not yet fully transformed all of its services to M-Government, thus SQUARE use could beneficial given its flexibility.

(2) Sampling criteria evaluates the extent of which SQUARE depends on sampling of the problem under inspection. We found that SQUARE relies on sampling; this was found evident in the second and third steps. The output of both steps are the goals, and artifacts which are critical and required as input to step four.

(3) Analyst permissibility criteria investigates how SQUARE supports an analyst's inputs. It was found that the analyst's inputs are of high value to SQUARE, where in each step of the process, the analyst has the freedom to choose the most suitable method; the one that outputs the most permissible results. As it was elaborated in the flexibility evaluation section, we were given the freedom to choose each method at each step of SQUARE.

(4) Interpretiveness evaluates whether SQUARE provides a detailed explanation of how to perform and complete each step of the process. By reviewing the official documentation of SQUARE, it was found that applying each step is simple; each step was explained given its goals, input type, techniques, participants, and the desired output. Furthermore, the official documentation of SQUARE provided a set of different techniques that the requirement team can choose from. Nevertheless, it was found that it does not provide a detailed explanation on how to perform each suggested technique, thus, the requirements team should search for the official document of the technique to understand its steps, or review a case study that was conducted by the selected technique.

(5) Data sufficiency criteria evaluates whether SQUARE supports the requirements team by providing step by step description for data gathering and analysis. It was found that SQUARE suggests the use of structured interviews, and focus groups to complete the first step of SQUARE. The official documentation of SQUARE suggested that the requirements team may seek additional support from public resources such as Software Engineering Body of Knowledge (SWEBOK) as a reference to complete the first step [24].

It was also found that SQUARE suggests the use of facilitated work sessions, surveys, or interviews to complete the second step while providing general guidelines on how to approach the task and align all the stakeholders' interest. SQUARE suggests a technique to be applied for a more

detailed brainstorming. That is to use mapping of mission-level availability requirements to system architectures and policy abstractions, which enables active mapping of high-level business requirements to low-level implementation requirements [25].

SQUARE provides an instruction of exit criteria when the desired output is complete or found. The data collection in this study was complete when there were no further major threats, vulnerabilities, nor security requirements to be found. It was found that SQUARE exhibits good data sufficiency, and that is due to its support to the process of data gathering and analysis.

(6) Coherence evaluates whether steps of SQUARE can be applied sequentially or in parallel. It was found that steps of independent activities that do not serve as an input for another step could be applied in parallel to save time. In contract, steps whose output serves as input to some other step(s) cannot be completed in parallel, thus they should be completed in a sequential manner. In more detail, steps one, two and three can be completed in parallel, while step four should be completed afterwards, and that is due to the fact that the output of steps two and three serves as input to step four.

VI. Conclusion

In this paper, an analysis of the SQUARE method was conducted for the purpose of allocating the requirements for M-Government transformation. The evaluation criteria of the SQUARE method were adaptability, complexity, implementation duration, and scalability. A description of each evaluation criteria was provided in the evaluation. Given the result of the last step of the SQUARE process, it was found that confidentiality, integrity, and availability are particularly important for the success of the M-Government transformation. Given the qualitative case study, we are unable to ensure the completeness of our results, as we focused primarily on the most important security threats, vulnerabilities, and requirements, which should be addressed in a future study. The future work should involve the use and evaluation of a systematic requirements engineering metamodel, e.g., [26], and a more in-depth analysis of security requirements at the architecture level [27]. The other directions might include the semi-automated SQUARE extensions development to reduce the analysis costs, e.g., [28]. We are also planning to evaluate SQUARE in the emerging blockchain domain [29].

References

[1] Telecommunication Regulatory Authority, UAE, "Mgovernment," https://goo.gl/2u6fdx, 2016, accessed: 2017-09-11.

[2] K. Schneider, E. Knauss, S. Houmb, S. Islam, and J. Jürjens, "Enhancing security requirements engineering by organizational learning," *Requirements Engineering*, vol. 17, no. 1, pp. 35–56, 2012.

[3] N. R. Mead and T. Stehney, "Security quality requirements engineering (SQUARE) methodology," *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 4, pp. 1–7, 2005.

[4] B. Fabian, S. Gürses, M. Heisel, T. Santen, and H. Schmidt, "A comparison of security requirements engineering methods," *Requirements Engineering*, vol. 15, no. 1, pp. 7–40, 2010.

[5] D. Mellado, E. Fernández-Medina, and M. Piattini, "A common criteria based security requirements engineering process for the development of secure information systems," *Computer Standards and Interfaces*, vol. 29, no. 2, pp. 244–253, 2007.

[6] B. Biel, T. Grill, and V. Gruhn, "Exploring the benefits of the combination of a software architecture analysis and a usability evaluation of a mobile application," *Journal of Systems and Software*, vol. 83, no. 11, pp. 2031–2044, 2010.

[7] A. Van Lamsweerde, "Elaborating security requirements by construction of intentional anti-models," in *Proceedings of the 26th IEEE International Conference on Software Engineering*, 2004, pp. 148–157.

[8] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, 2013.

[9] R. Kissel, "Glossary of key information security terms," NIST Interagency Reports, Tech. Rep. 7298-3, 2013.

[10] D. S. Government, "Dubai smart government mobile payment portal," https://mpay.dubai.ae/, 2016, accessed: 2017-09-11.

[11] S. K. Misra and N. Wickamasinghe, "Security of a mobile transaction: A trust model," *Electronic Commerce Research*, vol. 4, no. 4, pp. 359–372, 2004.

[12] G. Stoneburner, A. Goguen, and A. Feringa, "Risk management guide for information technology systems," https://goo.gl/Rtpoh, 2002, accessed: 2017-09-11.

[13] Joint Task Force Transformation Initiative, "Security and privacy controls for federal information systems and organizations," NIST Special Publication, Tech. Rep. 800-53, 2013.

[14] J. Brock, J. Boltz, E. Doring, and M. Gilmore, "Information security risk assessment practices of leading organizations," Accounting and Information Management Division, United States General Accounting Office, Tech. Rep. GAO/AIMD-00-33, 1999.

[15] S. A. Butler, "Security attribute evaluation method: a cost-benefit approach," in *Proceedings of the 24th international conference on Software engineering*, 2002, pp. 232–240.

[16] H. F. Lipson, N. R. Mead, and A. P. Moore, "A risk-management approach to the design of survivable COTS-based systems," Software Engineering Institute, Pittsburgh, PA, USA, Tech. Rep., 2001.

[17] A. P. Sage and Y. Y. Haimes, *Risk Modeling, Assessment, and Management*. John Wiley & Sons, 2015.

[18] S. L. Cornford, M. S. Feather, and K. A. Hicks, "DDP-A tool for life-cycle risk management," in *Proceedings of the 2001 IEEE Aerospace Conference*, vol. 1, 2001, pp. 441–451.

[19] R. P. Hubbard, "Design, implementation and evaluation of a process to structure the collection of software project requirements," Colorado Technical University, USA, Tech. Rep., 1999.

[20] J. Wood and D. Silver, *Joint Application Design: How to Design Quality Systems in 40% Less Time*. John Wiley & Sons, Inc., 1989.

[21] A. M. Davis, "The art of requirements triage," *Computer*, vol. 36, no. 3, pp. 42–49, 2003.

[22] B. Boehm, P. Grünbacher, and R. O. Briggs, "Developing groupware for requirements negotiation: Lessons learned," *IEEE Software*, vol. 18, no. 3, pp. 46–55, 2001.

[23] J. Karlsson and K. Ryan, "A cost-value approach for prioritizing requirements," *IEEE Software*, vol. 14, no. 5, pp. 67–74, 1997.

[24] P. Bourque, R. E. Fairley *et al.*, *Guide to the Software Engineering Body of Knowledge (SWEBOK) v3.0*. IEEE Computer Society, 2014.

[25] R. J. Watro and R. W. Shirey, "Mapping mission-level availability requirements to system architectures and policy abstractions," in *Proceedings of 2001 DARPA Information Survivability Conference and Exposition II*, vol. 1, 2001, pp. 189–199.

[26] H. Kaindl, M. Smiałek, D. Svetinovic, A. Ambroziewicz, J. Bojarski, W. Nowakowski, T. Straszak, H. Schwarz, D. Bildhauer, J. P. Brogan *et al.*, "Requirements specification language definition. project deliverable d2. 4.1, redseeds project (2007)," 2007.

[27] D. Svetinovic, "Architecture-level requirements specification." in *STRAW*, 2003, pp. 14–19.

[28] E. Casagrande, S. Woldeamlak, W. L. Woon, H. Zeineldin, and D. Svetinovic, "Nlp-kaos for systems goal elicitation: Smart metering system case study," *Software Engineering, IEEE Transactions on*, vol. 40, no. 10, pp. 941–956, 2014.

[29] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Transactions on Dependable and Secure Computing*, 2016.