Security Processes as Machines: A Case Study

Sabah Al-Fedaghi
Computer Engineering Department
Kuwait University
Kuwait
sabah.alfedaghi@ku.edu.kw

Manal Alsharah
IT-Services, Corporate Information Technology Group
Kuwait Oil Company
Kuwait
MHSharrah@kockw.com

ABSTRACT: *Business processes (called machines in this paper) are indispensable instruments for the realization of business activities of production and services. Business security processes (machines) are a type of machines whose role involves decreasing risks, responding to incidents, limiting exposure to liability and increase financial, physical, and personal risk values. Security machines are one of the most important nonfunctional business processes due to the possible effect of their failures for organizations in terms of finances, reputation and legal compliance. This paper focuses on capturing security as machines in a diagrammatic form either in the requirements phase of software development or as a necessary tool for documentation and communication in an ongoing system. A number of modeled security processes as machines are developed which can be integrated into business process streams in order to monitor different types of security aspects (e.g., confidentiality, integrity, availability). The paper purpose is to experiment with such a newly proposed machine-oriented approach to the notion of security process and develop case studies in actual business environments. The results points to the viability of the modeling methodology.*

Keywords: Conceptual Modeling, Diagrammatic Description, System Behavior, Process Control

Received: 8 January 2022, Revised 31 January 2022, Accepted 18 February 2022

DOI: 10.6025/dspaial/2022/1/2/49-61

1. Introduction

Business processes (will be called machines – defined precisely later in this paper) are indispensable instruments for the realization of their activities of production and services [1]. Business security processes (machines) are a type of machines whose role involves decreasing risks, responding to incidents, limiting exposure to liability and increase financial, physical, and personal risk values. Security machines are one of the most important nonfunctional business processes due to the possible effect of their failures for organizations in terms of finances, reputation and legal compliance [2].

Mitasiunas et al. [3] claim that "Security is a quality attribute of a system." Additionally, "security is a process oriented activity"

Mitasiunas et al. [3] (Italic added). That is, it is expressible in process oriented terms. In our language, a system is a machine and security is a sub-system consists of (abstract) machines as illustrated in Fig. 1. This thesis is not a new way of conceptualizing things (as will be defined later things are what can be created, processed, received, released and transferred)).

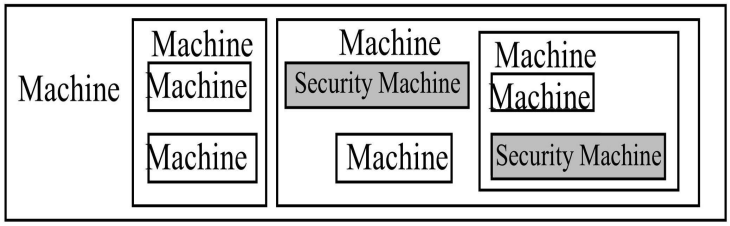

Figure 1. Security is a type of machine in the system

For example, the human body is made of 11 important organ systems (machines), including the circulatory, respiratory, digestive, excretory, nervous and endocrine, immune, integumentary, skeletal, muscle and reproductive systems. The immune system is security machines. According to Deleuze and Guattari [4], "the body (integrally with any mind that is involved in it) [is] a machine or swarm of machines.

The purpose of this paper is to experiment with such an machine-oriented approach to the notion of security process and develop case studies in actual business environments. The central idea for this methodology is based on a related work done in several publications [5-11]. Specifically, this paper focuses on capturing security as machines in a diagrammatic form either in the requirements phase of software development or as a necessary tool for documentation and communication in an ongoing system. A number of modeled security processes as machines are developed which can be integrated into business process streams in order to monitor different types of security aspects (e.g., confidentiality, integrity, availability).

1.1. Problem Discussed in this Paper

Several approaches to process modeling appear in the literature as well as in practice; e.g., software engineering, enterprise modeling, knowledge modeling, and workflow systems [12]. It is applied to all stages of the process life cycle, including analysis, design, enactment and control [13]. A business process consists of a chain of events, activities and decisions and involves a number of actors and objects [14].

Business process modeling is a means of representing the business activities, the information flow and decision logic in business processes. With the power of visualization, it is used to communicate information regarding a process and the interaction it includes within/between organizations either among the persons reading a model or the persons who create it. [15] (Italics added)

This paper emphases business process modeling that facilities human understanding and communication through creating an overall model for the design and development of the enterprise information system.

There is diversity of approaches to process modeling in this context such as Unified Modeling Language (UML) [16 Object][17] and event-driven process chains (EPC) [18].

"UML AD [Activity diagram] and BPMN are currently the two most expressive, easiest for integration with the interchange and execution level, and possibly the most influential in the near future. … The BPMN represents the high-level representation of business processes easily understood by business analysts and especially useful in communicating business requirements" [19].

This paper is applying a new diagrammatic language, called Flowthing Machine (FM) to business process modeling that has merits in certain aspects such as producing a uniform integrated representation based on one abstract machine that is applied levels. A process is a machine, a group of processes is a machine and the whole enterprise is a machine. Such a language could play a significant part as an initial step toward developing alternatives to the mixture of Havey's [20] modeling notations (e.g., BPMN + Petri nets [20]).

1.2. Problem and Current State of Modeling

This section gives several examples of diagrammatic modeling examples. The purpose is not to give a fair treatment of each of them, rather, the aim is to provide a view of the diagrams used in order to generally contrast them and attain some appreciation of the novelty of FM.

According to Argyropoulos et al [21], expert knowledge and security solutions can be captured in the form of process patterns, which can be integrated to business processes. They introduce process level security patterns, each of which contains the main activities required for different security requirements. Fig. 2 shows a sample of their modeling using BPMN collaboration diagram [22].

According to Saleem et al. [23], the general purpose modelling languages lack security elements, hence, they propose a domain specific language for security modelling along the business process modelling. Fig. 3 shows a sample of such a modelling representation.

Rodríguez et al. [24] focus on security requirements, in such a way that security is modelled along with the other aspects of a business process. Highly valuable requirements (including very abstract security requirements), are transformed into models with a lower abstraction level, such as analysis class diagrams and use case diagrams. They defined all the transformation rules necessary to obtain analysis class diagrams and use case diagrams. Fig. 4 shows the use cases used diagram used by Rodríguez et al. [24].

Morse et al [25] developed a system to allow encrypted data to be decrypted in a web-browser client and to re-encrypt it before being submitted to the server. This protects data to be stored without allowing support personnel on the server to use the data. Figure 5 shows their diagram that lays out how the encrypted data and the user-supplied passcode are used to decrypt the data.

To achieve a self-contained paper, the next section with reviewing the diagrammatic modeling tool FM. The example given in this section is a new contribution.

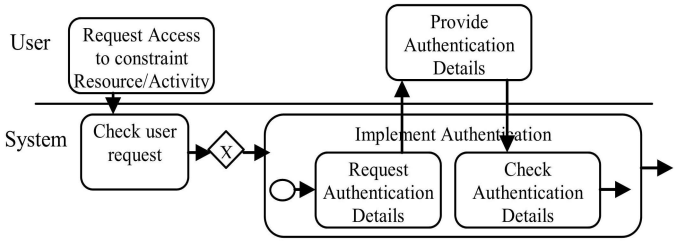

Figure 2. Authentication pattern (Re-drawn, partial from Argyropoulos et al [21]

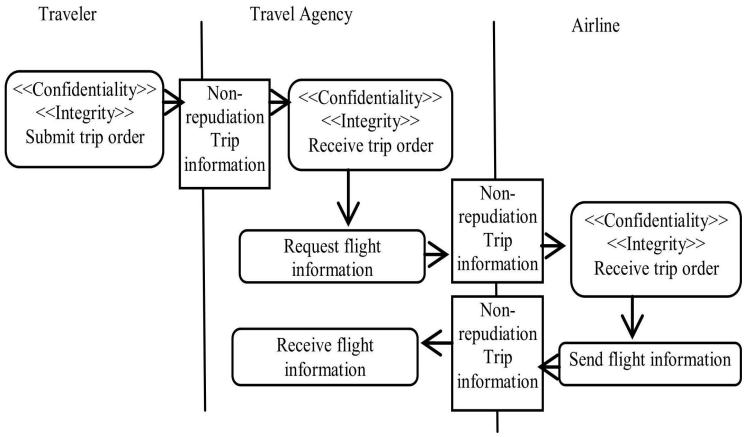

Figure 3. Security Enhanced Business Process Model (Re-drawn, partial from Saleem et al. [23]

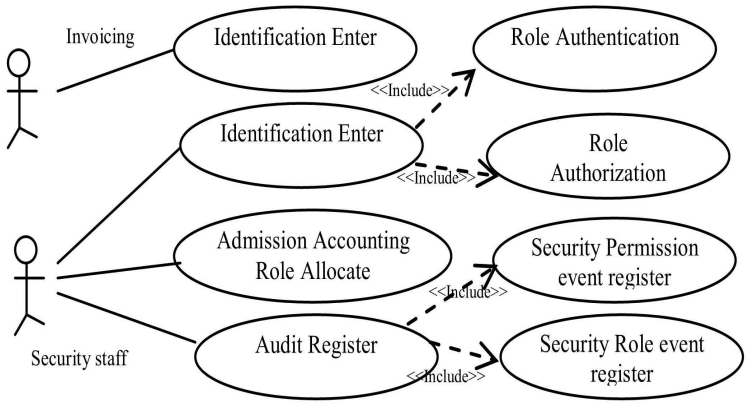

Figure 4. Access Control use case specification (Re-drawn, partial from Rodríguez et al. [24]

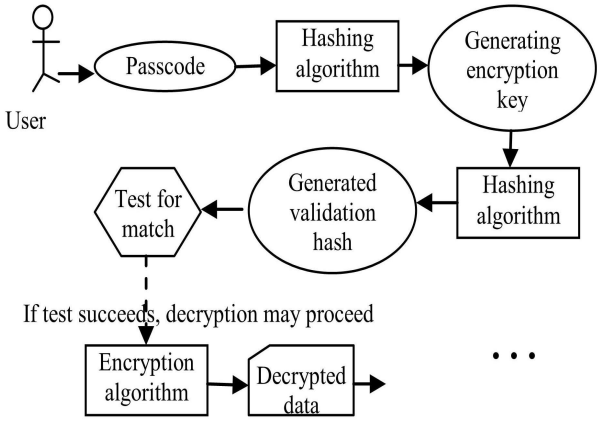

Figure 5. Encryption data flow (Re-drawn, partial from Morse et al [25])

2. Flowthing Machine Model

The Flowthing Machine (FM) model specifies conceptual flows into different stages of a system. This includes *things* that denote a range of physical and abstract items, including data, information, signals, objects, and events. Thing flow in an abstract machine (see Fig. 6) is based on six stages (states), as follows:

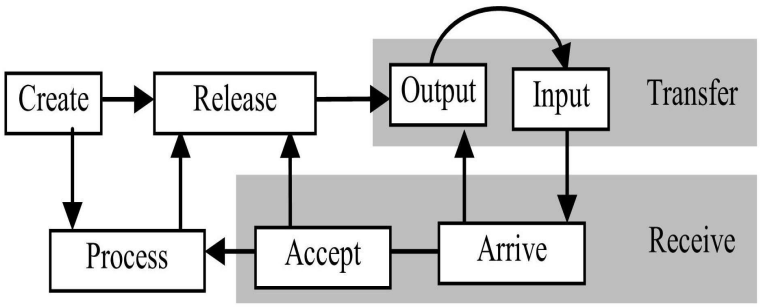

Figure 6. Flow machine

Arrive: A thing reaches a new flow machine (e.g., a thing arrives at a machine).

Accept: A thing is permitted to enter a machine; if arriving things are also always accepted, Arrive and Accept can be combined as a **Receive** stage.

Release: A thing is marked as ready to be transferred outside the flow machine.

Process (change): A thing goes through some kind of transformation that changes its form but not its identity.

Create: A new thing is born (created) in a machine.

Transfer: A thing is transported somewhere from/to outside the machine.

The stages in this machine are mutually exclusive (i.e., a thing in the Process stage cannot be in the Create stage or the Release stage at the same time). An additional stage of *Storage* can also be added to any machine to represent the storage of flowthings; however, storage is not an exclusive stage because there can be stored processed flowthings, stored created flowthings, etc.

A *thing* is defined as what can be created, released, transferred, arrived, accepted, or processed while flowing within and between machines. FM also uses the following notions:

Spheres and sub-spheres: These are the environments of the machine. Multiple machines can exist in a sphere if needed. A sphere can be an entity (e.g., a company, a customer), a location (a laboratory, a waiting room), a communication media.

Triggering: Triggering is the creation or activation of a flow by a point or condition in another flow (denoted in FM diagrams by a **dashed arrow**); e.g., a flow of electricity triggers a flow of heat.

3. Case Study

This section describes a conceptual model of an actual IT security process. To make this study more specific, we focus our discussion on the security processes related to *Digital Signature.* The architectural diagram (Fig. 7) and the description given by In a CGI Group Inc. [26]. It is entitled, *Key Encryption and Digital Signature: How do they work?* According to this document,

One of the major challenges facing consultants today is maintaining a level of knowledge of leading and emerging technologies, beyond the superficial or buzzword level. We need to develop a level of understanding that allows us to communicate effectively with both suppliers and customers… Beyond the infrastructure, there is a need for preparing, documenting and maintaining the infrastructure. [26]

Figures 8 and 9 are used to explain how PKI work.

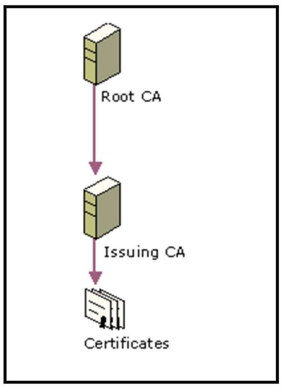

Figure 7. Architecture Diagram for the Case Study

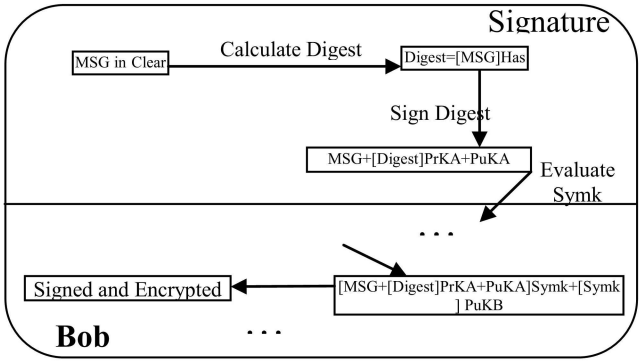

Figure 8. Signature and Encryption details with keys (Re-drawn, partial from CGI Group [26])

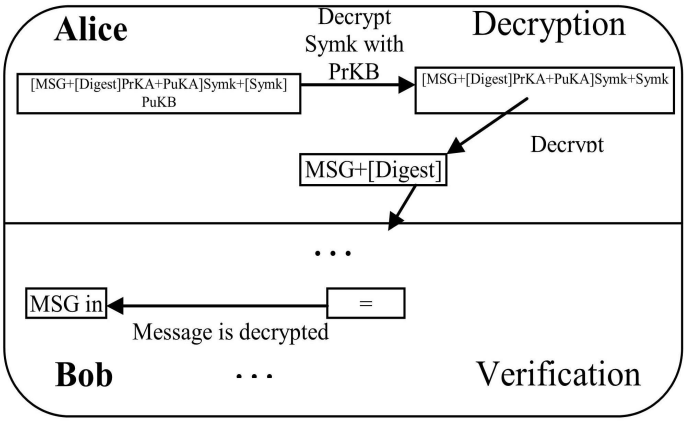

Figure 9. Decryption and verification details with keys (Re-drawn, partial from CGI Group [26])

FM provides an explicit diagrammatic description that can provide the documentation needed for these processed that can be used for understanding and communication among security experts and stock holders. It is richer and based on the simple notions of FM: the five basic stages, flow and triggering.

Accordingly, we build a model of the currently used specifications of the security processes related to Digital Signature.

3.1. Static Description (scenario1)

Fig. 10 is an FM representation of the process of issuing a secure digital certificate in PKI. The FM model shows how the end user can interact with the PKI system that consists of two servers: the root certificate authority (CA) server and issuing CA server.

In the figure, it is assumed that initially the CA has issued a self-signed digital certificate in which it contains its public and private keys (circle 1). The CA public key is available to all end users who are interacting with the system (2).

The user first registers to the system by sending a request (3). The request is received by the admin of the system who has a list of all the registered users (4). The admin processes the request (5) and makes sure that the user is authorized to use the system. If the user is not authorized, a rejection message is sent to the user (6). On the other hand, if the user is authorized to use the system, a confirmation email is sent (7).

All registered users can request a certificate (8). Once this request is created, it flows to root CA server in which it will be received by the authorization system and processed accordingly.

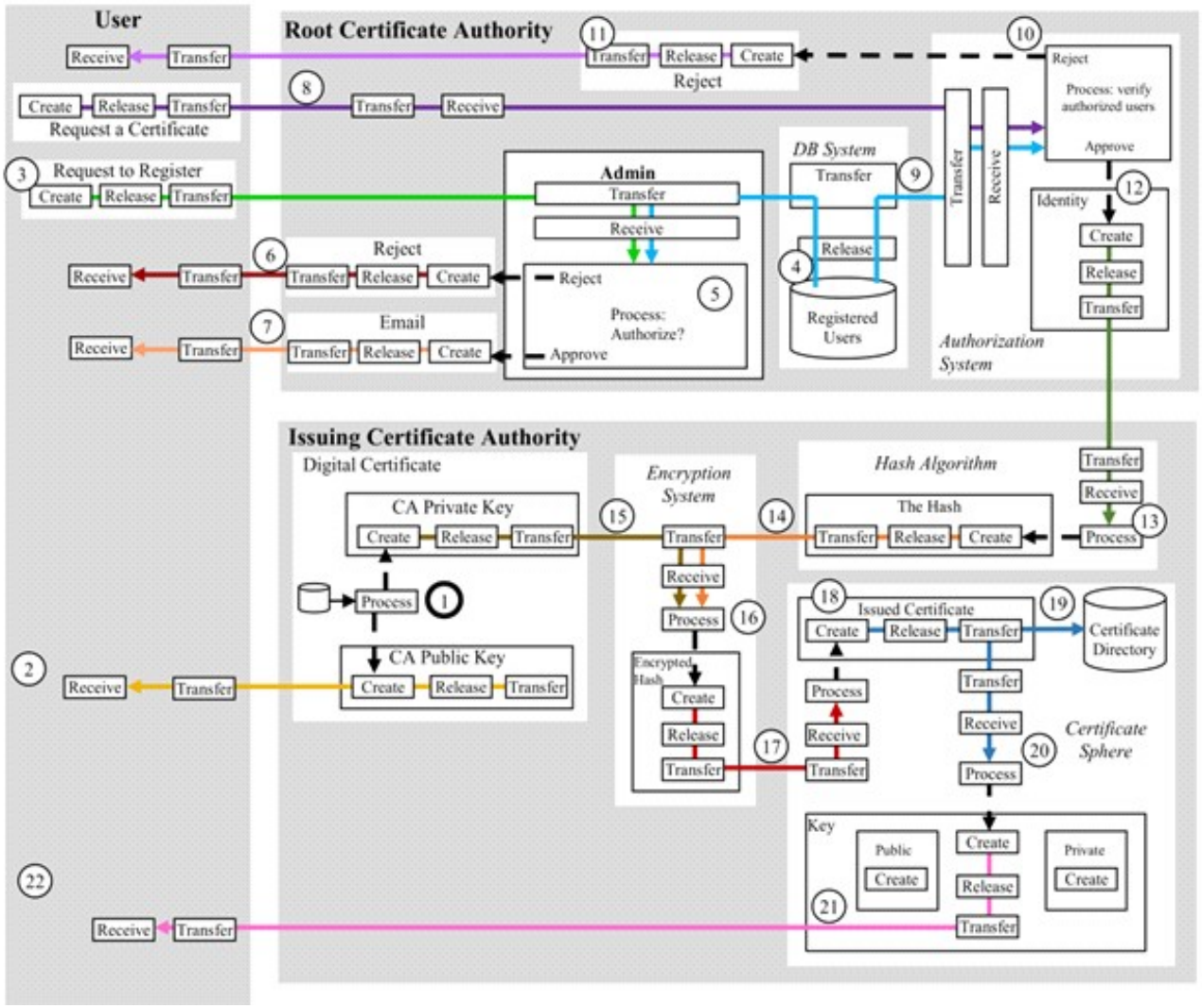

Figure 10. FM representation of the case study - Secure Issuing of Digital Certificates in Communication Using PKI (Scenario 1)

This system obtains the list of the registered users (9) to double check authorization since idle users are removed from the database if they registered and do not use their certificate for long time. The authorization system processes the request and,

• If it is rejected (10), a rejection message is sent (11).

• If it is approved, the system extracts and create the user's identity from the request (12) to compute a hash of the content that forms his/her certificate.

The identity is transferred to the issuing CA server where it is processed (13) to create the hash itself to be sent to the encryption system (14).

The hash is processed (16) by using the CA private key (15) that corresponds to the CA public key on the self-published CA certificate.

The encrypted hash is then created by concatenating (14) and (15). It is transferred (17) to make up the new certificate (18) that is stored in the certificate directory (19). The creation of the certificate triggers (20) the default creating of its public and private

keys (21) that makes the certificate content that is sent to end user (22).

3.2. Behavioral Specification (scenario 1)

Figure 10 reflects the static description of the system. To develop a specification of dynamic behavior, we identify the regions (sub-diagrams) of meaningful events, as illustrated in Figure 11.

Event 1 (E_1): The self-signed *digital certificate* is created by the system and the *CA public key* is available to the user.
Event 2 (E_2): A request to register is received by the system.

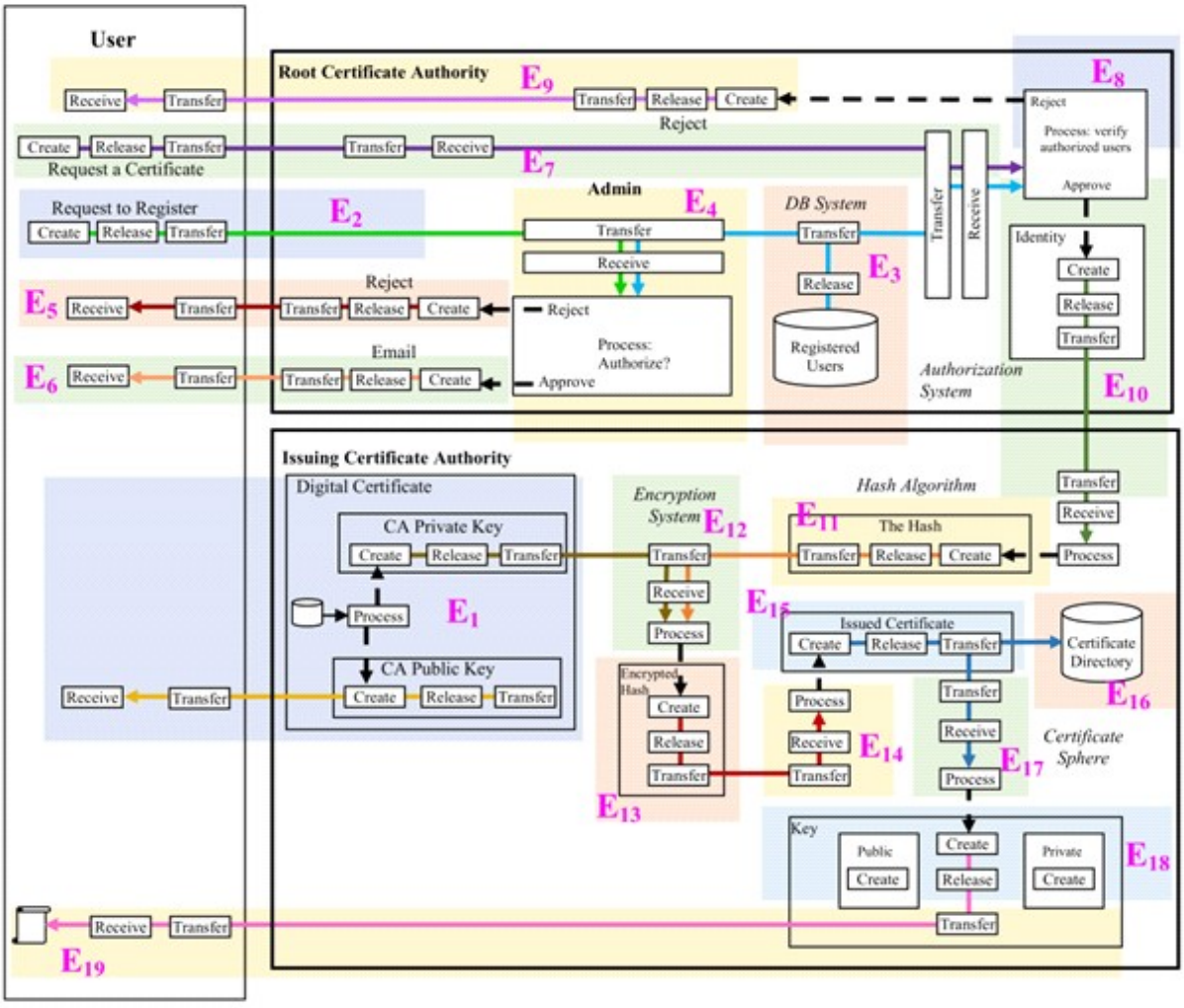

Figure 11. FM-Events Representation of the Case Study

Event 3 (E_3): The list of *registered users* is transferred to the admin.
Event 4 (E_4): The system admin processes the request to check for user's authorization.
Event 5 (E_5): Comparing the *initially registered users* with the request to register results in sending a rejection message.
Event 6 (E_6): Comparing the *initially registered users* with the request to register results in sending an approval email.
Event 7 (E_7): A request to have a new certificate is received by the system.
Event 8 (E_8): The user is verified for authorization.
Event 9 (E_9): Comparing the *initially registered users* with the new requester results in sending a rejection message.

Event 10 (E_{10}): Approved user that results in creating *Identity* to form the certificate.
Event 11 (E_{11}): *The Hash* is created.
Event 12 (E_{12}): The Encryption System concatenates *The Hash* with the *CA Private Key.*
Event 13 (E_{13}): The *Encrypted Hash* is created.
Event 14 (E_{14}): The *Encrypted Hash* is processed to form the certificate.
Event 15 (E_{15}): The *Issued Certificate* is created and transferred.
Event 16 (E_{16}): A copy of the *Issued Certificate* is saved in the Certificate Directory.
Event 17 (E_{17}): The *Issued Certificate* is processed to create its keys.
Event 18 (E_{18}): The *Public Key* of the certificate is created along with its corresponding *Private Key*.
Event 19 (E_{19}): The *Certificate* along with its *Keys* is transferred to the user.

Accordingly, these events can be represented in an event sequence diagram, Figure 12, to show the corresponding execution control

3.3. Sending & Receiving Message between Two Users using PKI: Static Description

This scenario focuses on the communication between two end uses that exchange messages using the PKI certificate to verify authentication. In this scenario, we assume that user 1 would like to send a secret message to user 2 over the corporate network. In addition, we assume that scenario 1 is already applied and that both users have received their PKI certificates. Fig 13 is an FM representation of this second scenario of our case study. First, user 1 requests the public key of user 2 (if it is not available in public) to use it in communication (1).

User 2 receive the request (2) and releases it (3) to flow over the network to user 1 (4). In constructing the message the Encryption System requires,

- The public key PKI (5).
- The original message (6).

The message is hashed (7) using any hashing algorithm to generate the hashed message (8) and flows to the PKI Encryption System (9). The PKI Encryption System encrypts the hashed message with user's 2 public key.

Additionally, in order to verify that the message has been written actually by the sender, user 1 signs the message by encrypting it using its own private key (10). The processing in PKI Encryption System processes (11) then triggers creating the signed encrypted message (12) that flows over the corporate network to be received by user 2 and reaches his/her PKI Decryption System (13).

In order for user 2 to read the received message, he/she uses his/her own private key (14) to decrypt it. In addition, to making sure that the message is actually coming from user 1 and it was not modified on its flow, the PKI Decryption System use user's 1 public key (15) to decrypt the message that was actually encrypted using user's 1 private key. The PKI Decryption System process the message (16) and triggers the generation of,

- The message itself (17) and
- The PKI Certificate (18)

Then the same hashing algorithm that was used (19) is used to create two hashes (20) and (21). Those hashes flow to the Verification System where the two hashes (22) are compared (23). If the two hashes matched, then the message is valid and the user can read it (24). If the two hashes are not matching, then the message is discarded (25) either because it was modified or it was corrupted.

It may be FM diagram look complex. This complexity is because it includes all conceptual specifications of stages, flow, and triggering. If the aim is just to show a general feature of the PKI process, a simplified diagram can be produced from Fig.13 most flow stages are eliminated as shown in Fig. 14. The figure highlights the main flows of the PKI process. This would be analogous

to representing the architectural drawing of a building (or a machine schema) with a simplified diagram of labeled geometric spaces. Figure 15 shows further simplification of the diagram. FM brings internal specifications up to the conceptual level. However, the FM details bridge the gap between expert and end user, analogous to building construction maps that are understood by owners as well as by different types of engineers.

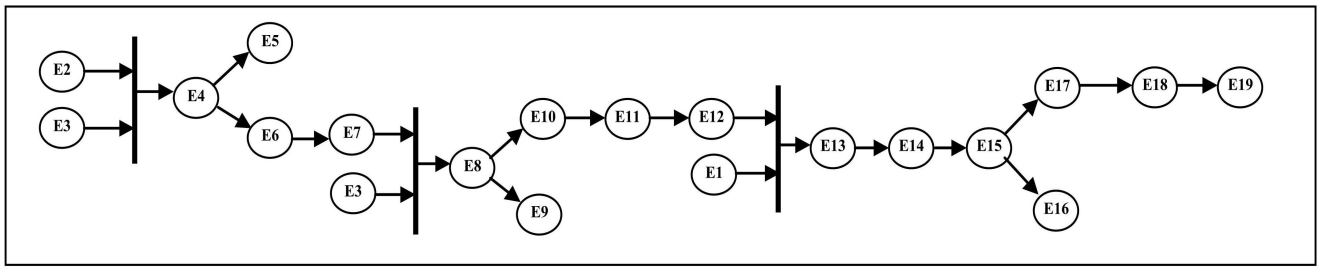

Figure 12. The Events Sequence

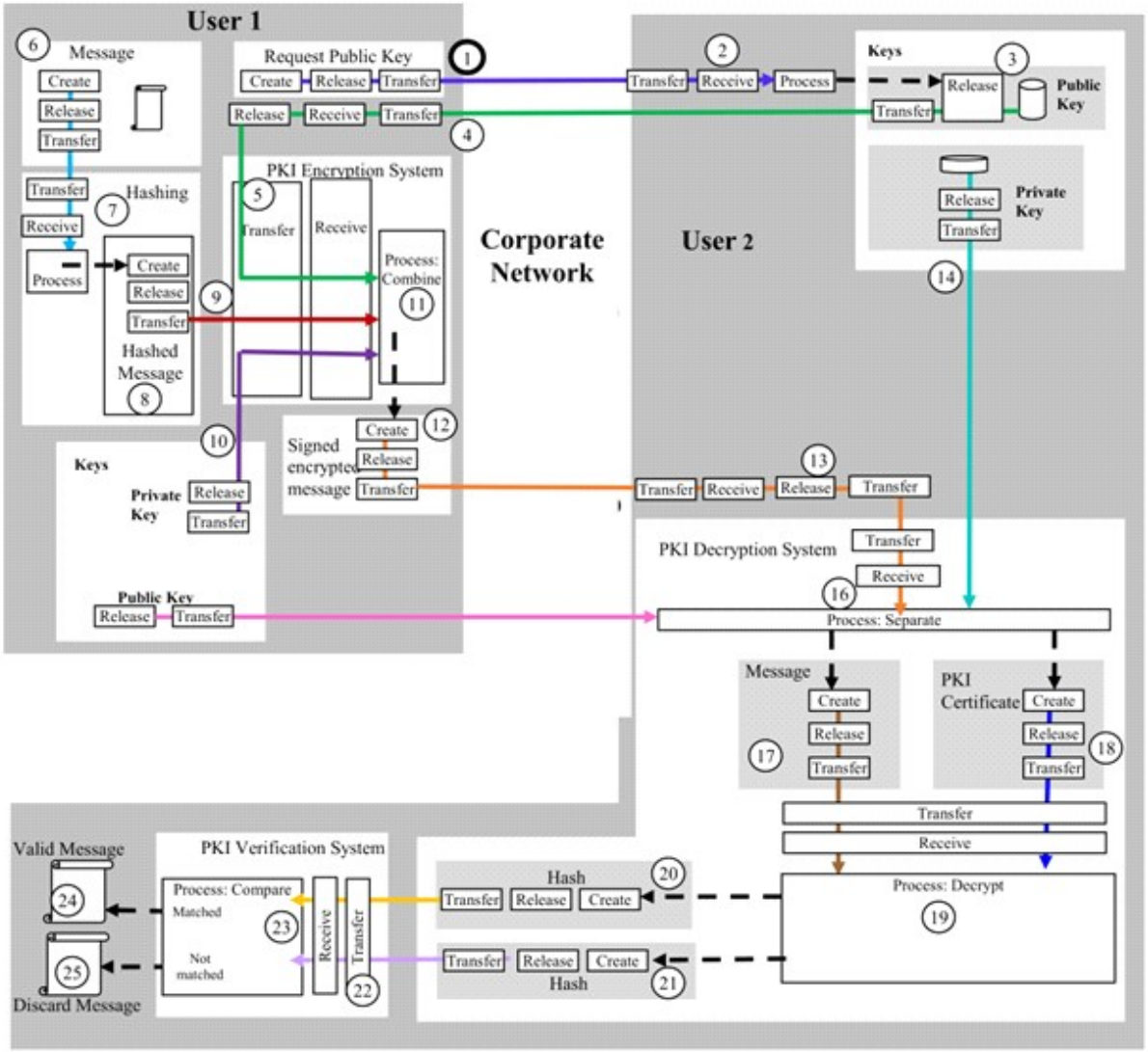

Figure 13. FM representation of the case study – Sending a Secrete Message between two Users (Scenario 2)

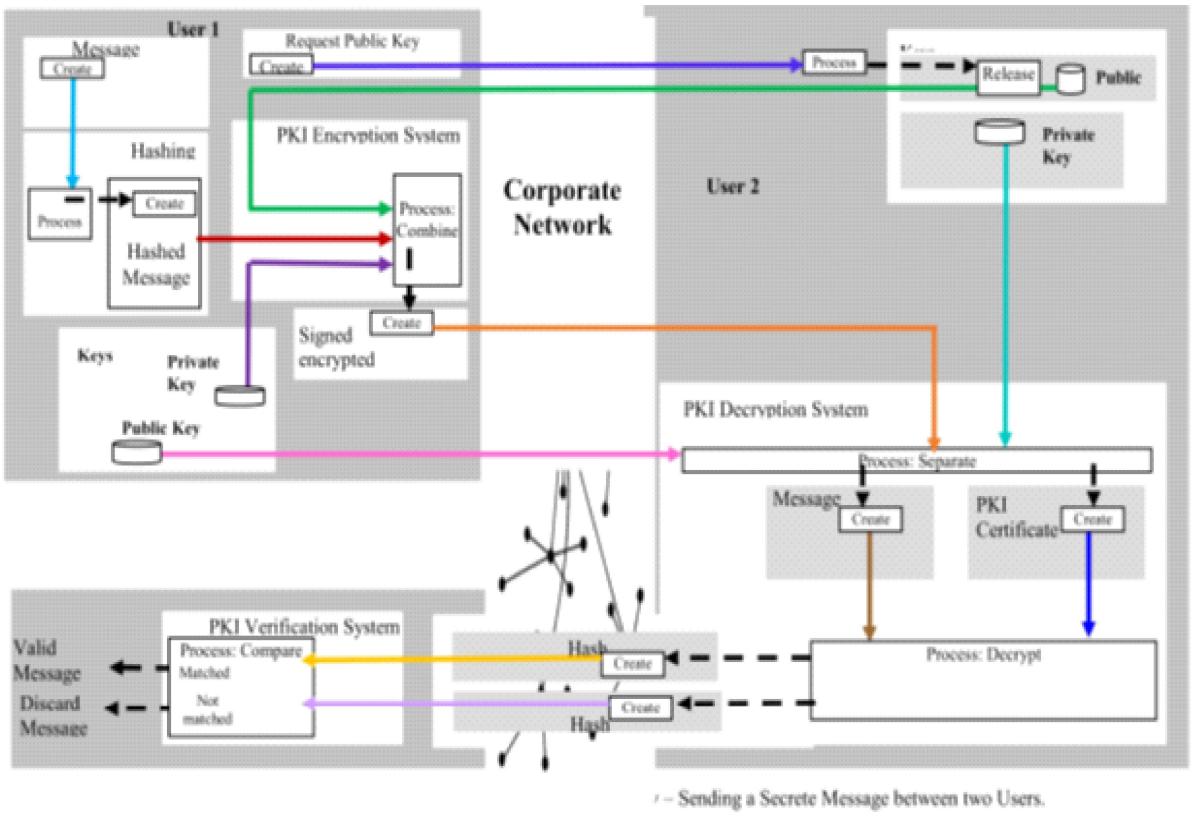

Figure 14. Simplification of the FM representation

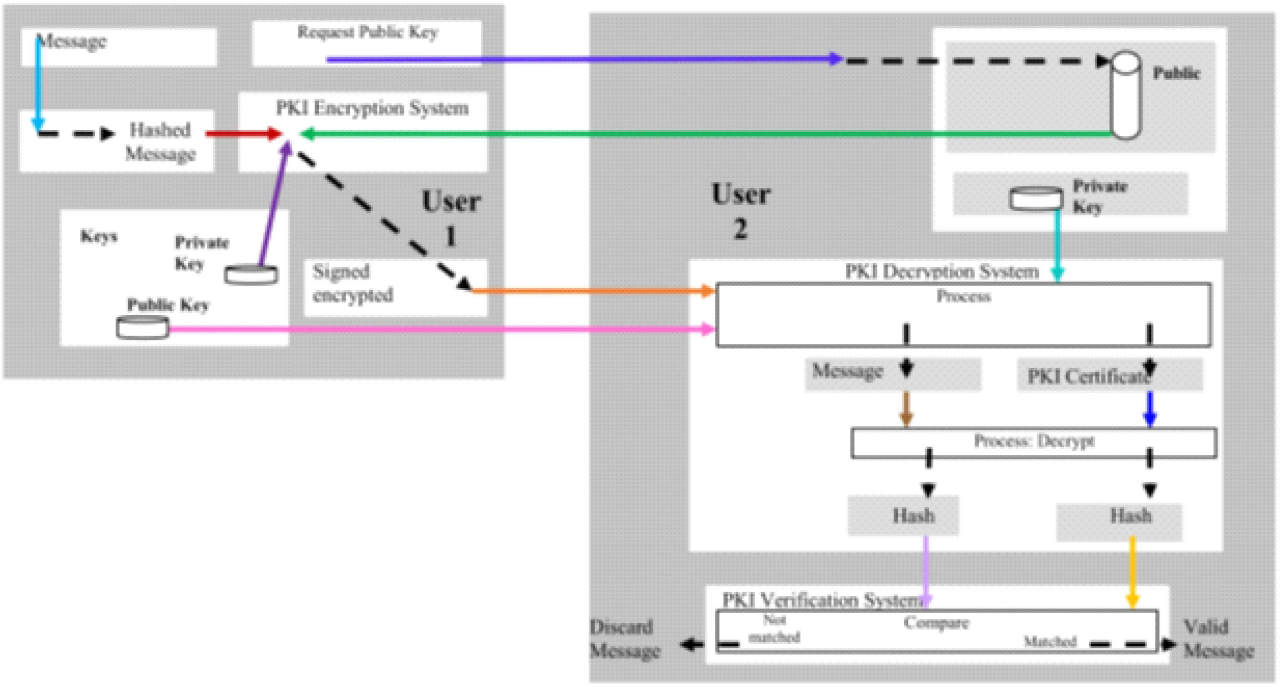

Figure 15. Further simplification of the FM representation of the case study – Sending a Secrete Message between two Users

The point from showing these diagrams is to emphasize that *details* are not complexity. In FM, if we are interested in less detail then it is difficult to reach the required level of specification. This is a familiar engineering practice as in electrical circuitry and architectural diagram where the model can be drawn at various levels of granularity. FM resolves the issue of complexity through simple, uniform notations applied across macro- and micro-levels of detail.

4. Conclusion

This paper has considered the process of generating abstract graphical models of security processes. The purpose is to experiment with such a newly proposed machine-oriented approach to the notion of security and develop case studies in actual business environments. The results point to the viability of the resultant high-level description seems to provide a tool that can be used in understanding, documentation, team communication, and teaching.

References

[1] Weske, M. Business Process Management: Concepts, Languages, Architectures.

[2] Neubauer, T., Klemen, M. & Biffl, S. (2006) Secure business process management: A roadmap. In: 1st International Conference on Availability, Reliability and Security (ARES 2006). *IEEE Publications.* Springer: Heidelberg (2010), p. 8.

[3] Mitasiunas, A., Novickis, L. & Kalpokas, R. (2014) Security process capability model based on ISO/IEC 15504 conformant enterprise SPICE. Applied Computer Systems, 15, Issue 1 (Jul 2014), 36–41

[4] Deleuze, G. & Guattari, F. (1983) Anti-Oedipus, translated from the French by R. Hurley. Marketer Seem, and Helen R. University of Minnesota Press: Lane (Originally 1972).

[5] Al-Fedaghi, S. & Alduwaisan, Y. (2018) Process modeling for analysis and control: A case study of IT services. *International Journal of Control and Automation,* 11, pp, 25–42.

[6] Al Fedaghi, S. & Abdullah, A. (2017) Flow-based systems modeling: Application in airport system description. International Journal of Industrial and Systems Engineering, 25, 318–334 [DOI: 10.1504/IJISE.2017.10002580].

[7] Al-Fedaghi, S. (2017) Context-aware software systems: Toward a diagrammatic modeling foundation. *J. Theoretical Appl. Inform. Technol.*, 95.

[8] Al-Fedaghi, S. & Alahmad, H. Orientation in conceptual modeling frameworks 3rd IEEE International Conference on Big Data Intelligence and Computing, Orlando, USA, 6–10 November, 2017 [DOI: 10.1109/DASC-PICom-DataCom-CyberSciTec.2017.209].

[9] Al-Fedaghi, S. & Alahmad, H. (2018) Integrated modeling methodologies and languages. *Academic Medicine 12th International Conference on Ubiquitous Information Management and Communication*, Langkawi, Malaysia, 5–7 January, 2018.

[10] Al-Fedaghi, S. Activity recognition and sensor positioning *IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2016)*, 9–12 October, Budapest, Hungary.

[11] Al-Fedaghi, S. Schematizing proofs based on flow of truth values in logic IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2013), 13–16 October, Manchester, UK [DOI: 10.1109/SMC.2013.40].

[12] Aagesen, G. & Krogstie, J. (2010), pp Analysis and design of business processes using BPMN. In: Handbook on Business Process Management 1. International Handbooks on Information Systems (edited by J. vom Brocke & M. Rosemann). Springer-Verlag: Berlin.

[13] von Rosing, M., Foldage, U., Hove, M., von Scheel, J. & Bøgebjerg, A.F. (2015). *Working with the Business Process Management (BPM) Life Cycle, the Complete Business Process Handbook*, Vol. 1. Elsevier: Amsterdam [DOI: 10.1016/B978-0-12-799959-3.00014-8].

[14] Dumas, M., La Rosa, M., Mendling, J. & Reijers, H. (2018). Fundamentals of Business Process Management. Springer-Verlag: Berlin, Heidelberg [DOI: 10.1007/978-3-662-56509-4].

[15] Lampathaki, F., Koussouris, S. & Psarras, J. (2013) *Business process modeling: Business process reengineering*. Location.

[16] Object Management Group. *About the Unified Modeling Language Specification*, version 2.5 (2015). www.omg.org/spec/UML/About-UML/.

[17] Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F. & Vandin, A. (2017) BProVe: A formal verification framework for business process models. *In: Proceedings of the* 2017 32nd *IEEE/ACM International Conference on Automated Software Engineering* (ASE). *IEEE Publications*, pp. 217–228.

[18] Scheer, I.D.S. & ARIS (Architecture of Integrated Information Systems) Location. Springer: Berlin (1992).

[19] Ko, R.K.L., Lee, S.S.G. & Lee, E.W. (2009) Business process management (BPM) standards: A survey. Business Process Management Journal, 15, pp, 744–791 [DOI: 10.1108/14637150910987937].

[20] Havey, M. (2006) Keeping BPM simple for business users: Power users beware. BPTrends. mchavey.blogspot.com/2016/04/the-mike-havey-collection.html#. January 2006!/2016/04/the-mike-havey-collection.html.

[21] Argyropoulos, N., Mouratidis, H. & Fish, A. (2017) Supporting secure business process design via security process patterns. In: BPMDS/EMMSAD 2017, LNBIP, Vol. 287 (edited by I. Reinhartz-Berger), pp. 19–33.

[22] Specification of Business Process Modeling Notation version 2.0 (BPMN 2.0) [Online] (2017). www.omg.org/spec/BPMN/2.0/PDF.

[23] Saleem, M.Q., Jaafar, J. & Hassan, M.F. (2012) Security modelling along business process model of SOA systems using modified “”UML-SOA-Sec” International Conference on Computer & Information Science (ICCIS).

[24] Rodríguez, A., Guzmán, I.Gd, Fernández-Medina, E. & Piattini, M. (2010) Semi-formal transformation of secure business processes into analysis class and use case models: An MDA approach. *Information and Software Technology*, 52, 945–971.

[25] Morse, R.E., Nadkarni, P., Schoenfeld, D.A. & Finkelstein, D.M. (2011). www.biomedcentral.com/1472-6947/11/70 *Web-browser encryption of personal health information. BMC Medical Informatics and Decision Making*, 11, 70.

[26] CGI Group, Inc. (2004). Public Key Encryption and Digital Signature: How Do They Work? [White paper]. https://www.google.com/url?sa= t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiGgOiR1qjaAhWGtBQKHX5CCvYQFggkMAA&url=https%3A%2F%2Fwww.cgi.com%2Ffiles%2Fwhite-papers%2Fcgi_whpr_35_pki_e.pdf&usg=AOvVaw1XscgpmO3SSPLFg30LoE4G.