# Towards Detecting and Mitigating Conflicts for Privacy and Security Requirements

Duaa Alkubaisy, Karl Cox, Haralambos Mouratidis

CSIUS (Centre for Secure, Intelligent and Usable Systems) University of Brighton

Brighton, United Kingdom

{D.alkubaisy, K.Cox, H.Mouratidis}@brighton.ac.uk

***Abstract*—requirement engineering live in a world were contradiction is the norm. Hence, development of software engineering is usually an adjustable and upgrading cyclical process. We found in the literature that some requirements conflict with other requirements. We will focus in this study on identification and resolution of conflicts between security and privacy requirements. Although, most the recent studies focus on identifying conflicts without proposing a solution to resolve it. This paper presents an approach to identifying and resolving conflicting privacy and security requirements as patterns. By using patterns to describe the problem we can propose a solution for each conflict.**

***Keywords*— *software engineering; security requirements; privacy requirements; conflict; strategy of resolution.***

## I. INTRODUCTION

Information systems are used everywhere and are becoming more complex. One aspect adding to that complexity is the growing number of security and privacy requirements that need to be built into the product. Moreover, with the new **GDPR** regulation (General Data Protection Regulation), agreed upon by the European Parliament and Council in April 2016, replacing the Data Protection Directive in spring 2018 as the primary law regulating how companies protect EU citizens' personal data needs a conclusion to this point [1] and regulation to protect privacy, since information system holds personal confidential information. Privacy is designed in software information systems. But this sometimes creates issues: on one hand we have security requirements and other hand we have privacy requirements, and these likely conflict.

The challenge is the literature doesn't appear to provide adequate methods to identify and mitigate conflicts between security and privacy requirements. However, identifying and mitigating such conflict is important to reduce risk on the software system, by having conflicts at least one of the security or privacy requirements will be vulnerable and easy to be target.

Conflicting requirements are identified as one of the three primary causes of additional effort or mistakes during software development [2]. Conflict can be defined as a clash of interest between two aspects of development: security and privacy. These conflicts may occur within a level of the project, such as among items at the goal level, requirements level, technical level, or implementation level.

## II. Motivating

To represent the problem statement we can find that this issue are very common, almost in each sector, for instance, banking, education, and health care. All those sectors required to insure users' privacy, in the meantime to maintain system secure and invulnerable.

In order to have a better understanding of the problem, in case of having conflicts between requirements. We inspired by Q, Ramadan et al. [3] E-Health case study. Because it shown exactly how we identify conflicts between requirements. By represent the model. In the E-Health usually patients have strong privacy concerns about how and for what reason their health information is handled, which could cause an interfere with an organization's documentation responsibilities to certifying accountability.in other words patients priority to keep their information anonymity and undetectable, and that would conflict organisation goals such as accountability. Moreover, the model explain how a patient can use a telemedicine device in order to receive an over-distance healthcare service. Moreover a patient can also evaluate the service through an online evaluation portal. By having an example like E-Health system its appear to us many requirements could be conflicting, hence this paper focused on security and privacy requirements we will describe phase by phase how to Identify conflict between security and privacy requirements toward resolve or mitigate conflicting between requirements.

## III. Related Work

Various studies have been conducted regarding the decision criteria to use in non-functional requirements conflicts. There is an interesting study that identifies conflicts between security and data-minimization requirements is a challenging task. Since such conflicts arise in the specific context of how the technical and

978-1-7281-4844-1/19/$31.00 ©2019 IEEE

organizational components of the target system interact with each other, their detection requires a thorough understanding of the underlying business processes. To address this challenge, Ramadan *et al.* [3] propose an extension of the business process modelling language (BPMN modeling language) to enable specification of process-oriented data-minimization and security requirements, also to detect conflicts between these requirements based on a catalogue of domain-independent anti-patterns. However this study is relevant to our work, it differs in the level of business process modelling, while our approach focused in a different SDLC stage (requirement engineering phase). In addition, Ramadan *et al.* do not consider resolving conflicts as part of their approach. Though Ramadan *et al* study is information rich that helps us in identifying conflicts and mapping the most requirements being conflicting.

In addition, Egyed and Grunbacher [4] used an automated traceability technique to eliminate false conflicts and cooperation. Analyzing the requirements is the first step and then an identification of the requirements is made based on their attributes which are cooperative or conflicting. Then, the trace analyzer automatically detects the trace dependencies among the requirements. The system aids in determining the extent requirements overlap by using trade dependencies knowledge. If two requirements overlap, then the two requirements are conflicts. Whereas, if there is no overlap between them, then conflicts cannot exist. Another technique used to identify the conflicts is known as the negotiation method. Here, the stakeholders and software engineers discuss the project orally and analyze the requirements as well as the conflicts that the project could be facing [5].

Mairiza and Zowghi [6] suggest that to manage the conflicts, some techniques can be used for viewing, interpreting and evaluating NFRs. It is important to assess NFRs while knowing their impact and importance to the system. Mairiza et al. [7] applied an experimental approach to design a framework that manages the relative conflicts among NFRs. As well, Santana and Liu [8] proposed a framework to analyze the conflicts among non-functional requirements using the integrated analysis of functional and non-functional requirements. The conflict detection is performed on high-level NFRs based on the relationship between quality attributes, constraints and functionality.

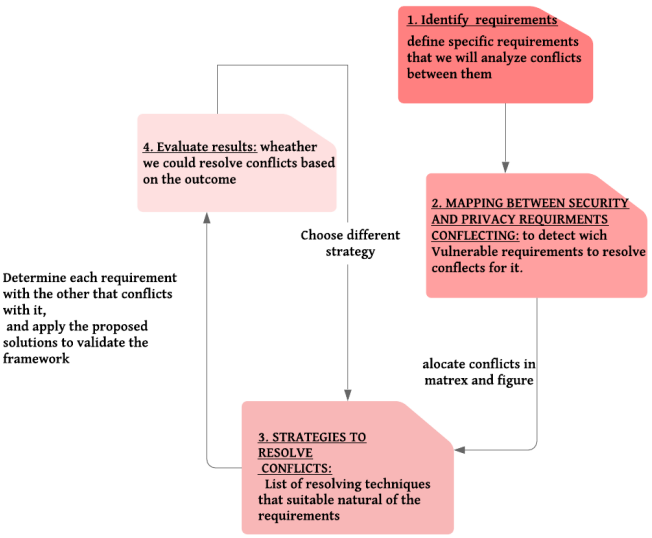
From the recent studies, we can identify that the following research gaps in the area of managing security and privacy requirements frameworks. That first of all, normally frameworks consider either security or privacy issues, we find very limited frameworks focused on both security and privacy. In addition, we find that most of approaches focused on identifying or analyzing conflicts as we found it's so infrequent to find a study consider resolving conflicts between the requirements.

In this research, we will present a way to enclose security and privacy requirements, after that will map between those conflicting requirements, to find which requirement is more vulnerable. We then apply the suitable resolution techniques being described in the third section of the paper.

## IV. FRAMEWORK

The approach aims to achieve two goals: to identify a conflict between security and privacy requirements and to find a resolution technique to resolve or reduce different issues. The process begins with elicit requirements that we will apply it to the approach. Once the requirements are specified, the next step is listing conflicts between specific security and privacy items. Strategies to resolve the listed disputes must then be used. A common method is to describe a problem with an example, propose a solution that keeps the requirement in mind, and apply this solution to the case.

**FIGURE1: FRAMEWORK PPROCESS**




### A. *IDENTIFY REQUIRMENTS*

The first step of detecting conflicts is to review the literature to find out more about conflicting issue provide it with some examples to detect how this conflict effect the system. Below are the most frequent requirements in the security and privacy aspects of software engineering [9].

#### *1) Security Requirements:*

**Authentication** is the task of determining whether an entity is what or who it is declared to be. This process involves the validation of identity. **Authorization** is the next logical step, wherein the identified entity is checked for the necessary permission or privileges to access data and resources. **Confidentiality** allows for the limitation of access to or exposure of a specific resource as dictated by policy [10].

| SECURITY REQUIREMENTS | PRIVACY REQUIREMENTS |
|---|---|
| AVAILABILITY | Anonymity |
| NON REPUDIATION | Unlinkability |
| CONFIDENTIALITY | Pseudonymity |
| INTEGRITY | Unobservability |
| AUTHENTICATION | Unlinkability |
| AUTHORIZATION | Undetectability |
| SEPARATION OF DUTIES (SOD) | |
| BINDING OF DUTIES (BOD) | |
| ACCOUNTABILITY | |
| AUDITABILITY | |

**Table1: elicit most frequent the security and privacy requirements being conflict**

**Non-repudiation** ensures that a user or entity provides an undeniable signature that accounts for their actions. **Integrity** is responsible for ensuring that all information is accurate and consistent during its use and has not been altered unknowingly or otherwise. **Availability** assures that data is always accessible and can easily be provided to an authorized entity. Denying information can cause both inconvenience and delays,which may prove to be critical. **Separation of Duties** acts as a restricting agent for any individual to have too much or inappropriate control over the system. **Binding of Duties** similarly ensures that two separate entities are needed to have sufficient control over the system. **Accountability** is the requirement that holds entities responsible for their actions or lack thereof. **Auditability** ensures that a trace can be done on an entity's activities within the system. [11].

### *2) Privacy Requirements:*

Privacy requirements are often in compliance with existing data laws or rules within a country. For a project to be compliant, they must be able to ensure privacy within the system. [12].

First, **Anonymity** allows entities to use resources or services without having to reveal their identity. **Unlinkability** ensures that an entity can use a service without being associated with the service itself. **Pseudonymity** gives the users the freedom to work under an alias or aliases, without having to provide personal information sufficient to determine their identity. **Unobservability** denies any entity from knowing for sure that a user is accessing a service, as well as the inability to track a user's actions while using a service or resource. **Undetectability** ensures that an entity cannot identify which user among a user pool is accessing the service. [13] With this, it is apparent that some aspects of security and privacy requirements are already in conflict with each other. Security requirements, such as **Accountability**, **Authenticity**, **Auditability**, and **Non-repudiation**, require a log of movement and activity within the system. However, these are directly in conflict with the privacy requirements of **Anonymity** and **Unobservability**, which should conceal the user's actions. **Binding of duties** and **Separation of duties** could conflict with **Anonymity** and **Unlinkability** as well since the steps to be executed would have to verify their identity. Other aspects may have conflicts which are not apparent at the requirements stage. For example, the security requirements **Confidentiality**, **Integrity**, and **Availability** depend on having proper authorization to access or modify resources. Identification is not necessary to achieve these requirements, but it is a common approach which may be used by the system developers. As such, it opens a potential conflict between the security requirements mentioned above and data minimization privacy requirements. Conflicts can also occur within each aspect, whether of security or privacy. These conflicts arise when more concrete requirements are specified. For example, if a user should be required to access a service using their alias, then it conflicts with the general concept of **Anonymity**. However, if some aspects supplement or overlap concerning requirements, then they cannot be considered as conflicts [14]. **Confidentiality**, **Integrity**, and **Anonymity**, which are different aspects but ultimately strive towards the same goal of protecting data against unauthorized tampering, do not conflict with each other.

## *B. MAPPING BETWEEN SECURITY AND PRIVACY REQUIRMENTS CONFLECTING*

The matrix maps conflicts between security requirements and privacy requirements. While there may indeed be conflicts among security requirements themselves, this will focus on conflicts that cross the two aspects. The matrix helps to visualize the requirements with the most conflicts, which aids in identifying which ones deserve focus. From this matrix, **Anonymity** and **Unobservability** conflict the most with other security requirements. In order to have more visual mapping to point on most frequent requirements having conflicts. Based on our previous studies we found that those five security requirements are luckily being conflict with more than one privacy requirements. As it shown below

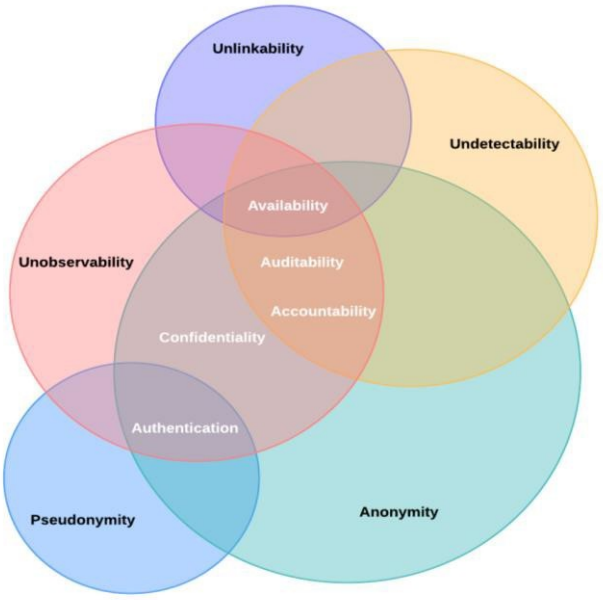



**Figure2: detect conflicts between security and privacy requirements**

| | | Security Requirements: | | | | | | | | | | Privacy Requirements: | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Availability | Non repudiation | Confidentiality | Integrity | Authentication | Authorization | SOD | BOD | Accountability | Auditability | Anonymity | Unlinkability | Pseudonymity | Unobservability | Undetectability |
| Security Requirements: | Availability | | | | | | | ○ | | | | ○ | ○ | | ○ | ○ |
| | Non repudiation | | | | | | | | | | | ○ | | | ○ | |
| | Confidentiality | | | | | | | | | | | ○ | | | ○ | |
| | Integrity | | | | | | | | | | | ○ | ○ | | ○ | |
| | Authentication | | | | | | ○ | | | | | ○ | | ○ | ○ | |
| | Authorization | | | | | ○ | | ○ | | | | | | | ○ | |
| | SOD | ○ | | | | | ○ | | | | | | | | | |
| | BOD | | | | | | | | | | | | ○ | ○ | | |
| | Accountability | | | | | | | | | | | ○ | | | ○ | ○ |
| | Auditability | | | | | | | | | | | ○ | | | ○ | ○ |
| Privacy Requirements | Anonymity | ○ | ○ | ○ | ○ | ○ | | | | ○ | ○ | ○ | ○ | ○ | ○ | |
| | Unlinkability | ○ | | | ○ | | | | ○ | | | ○ | | ○ | | ○ |
| | Pseudonymity | | | | | ○ | | | ○ | | | ○ | ○ | | | |
| | Unobservability | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | ○ | | | | ○ |
| | Undetectability | ○ | | | | | | | | ○ | ○ | | ○ | | ○ | |

**Table2: mapping conflicts between security and privacy requirements**

As it appear to us at figure 2 that **Availability** as a security requirement is more complicated requirement while it involved with most privacy requirements therefore it apparently being conflicts with four privacy requirements: **Anonymity, Unlinkability**, **Unobservability** and **Undetectability.** Following we found that **Confidentiality, Accountability and Auditability** a security requirements could be conflict with three of privacy requirements. Finally **Authentication** security requirement always involved with two of privacy requirements (**Anonymity and Pseudonymity**) as we mentioned in the example in the introduction part.

## C. *STRATEGIES TO RESOLVE CONFLICTS*

***Conflict Resolution:*** is comprised of conflict management and negotiation. The former aspect is a process that after identifying conflicts between requirements, it could be a prioritization or negotiation. To have the right decision of which requirement the framework will support. We will describe further about those strategies and criteria to conceder to choose the suitable strategy in a next paper. In this part we enclose some common tools to support requirements, some tool can work with more than one requirement, while some tools conduct for a specific requirement. We will have a brief idea about each tool, after that we will classified them base on security, privacy requirements or maybe both!

***Tools:*** The most requirement being conflict is Availability, therefore we could use Redundancy to the system to secure the information. Redundancy, or the duplication of critical points of the system, ensures that an application is reliable and available for its intended users. Should a function or component fail, another instance would be ready to take its place so that the system performs with little or no downtime [15].

However, a disadvantage is that redundancy increases both cost and complexity of the system. An architecture that correctly models the system is vital to the success of high availability. While redundancy is not necessary in many applications, it is a critical component if system failure or downtime has severe consequences.

Cryptography encryption of data is a common way of security of implementing confidentiality [16]. Standard examples include simple passwords, security tokens, and two-factor authentication. Both symmetric and asymmetric algorithms may be used to provide encryption. This kind of encryption works for security requirements (Confidentiality). For Integrity we have Message Authentication Codes (MAC): A fundamental technique to verify both the integrity and authenticity of transmitted data are Message Authentication Codes (MACs). Initially, the construction of most of these codes were based on pseudorandom functions which were either through fast block-cipher based algorithms or slower number-based theories.. Dummy Traffic occurs when a fake message is included in a mix network with the goal of confusing an attacker. With a dummy message, passive and active attacks on a network are more difficult, thus creating a more secure system [17]. Mixes usually generate these dummy messages, allowing for a higher level of anonymity and prevention of end-to-end intersection attacks. This method generally could apply on most privacy requirement except Pseudonymity. In the other hand we have some resolution techniques to protect privacy requirements: Third-party Conflict Resolution, is the most common method enlists the help of a neutral third party for management or resolution of a dispute. With the unbiased opinion of a third party [18].conflict resolution can be made in a calm and organized manner. This resolution also ensures that working relationships are somewhat preserved as a mediator is in place. This method is suitable for Anonymity, Unlinkability and Undetectability. Zero-knowledge Protocol allows a

party to prove that a statement is certainly true without revealing additional information [19]. This protocol must have the following properties: Completeness, that the honest verifier should be convinced by an equally honest prover; Soundness, which the probability of satisfying a verifier that a false statement is true is minimal; and Zero-knowledge, that a cheating verifier can learn nothing from the statement but the truth. Usually this method mechanism for Anonymity.

Trade-off Analysis is a simple give-and-take wherein one quality, quantity, or property is lost or diminished to increase these in another aspect. Trade-offs are usually obtained through discussions and sharing of insights. [20].Toward resolve conflicts for Pseudonymity requirements we should use Public key [21].

Pseudonymity provides a consistent identity without having to tie it to a specific physical person or organization. It allows for the advantages of having a known identity, such as accountability, while still maintaining anonymity. One way to implement pseudonymity is through a public key that verifies digital signatures anonymously made by the holder of the corresponding private key. Users can create their own public keys for digital pseudonyms. Each key pair may be bound to an email address, self-certified, and used thereafter. Moreover, Steganographic technologies to resolve conflicts in Unlinkability and Undetectability requirements. Stenography is the art of invisible communication, a technique where data is transmitted in a way that conceals the existence of another message. [22] Unlike cryptography, which only encrypts the message itself, stenography encrypts the message such that unauthorized parties would not be aware of a message at all, this method works well because a change in the least significant bit (0 to 1 or vice versa) does not drastically change the overall appearance of the image.

**We categorize those tools base on:**

1- Techniques are suitable for both security and privacy requirements, that's mean that one technique could support either security or privacy requirements.

2- Some techniques support as for privacy requirements only or security requirements. This decision based on a previous step (requirement prioritization) to know which requirements the analyst will choose to support.

**1-Techniques suitable for either Security or Privacy requirements**

| SECURITY AND PRIVACY REQUIREMENTS | TOOL TO SUPPORT REQUIRMENT |
|---|---|
| **Anonymity VS Confidentiality** | Cryptographic, Steganographic technologies, Onion routing |
| **Unlinkability VS Confidentiality** | Cryptographic, Steganographic technologies, Homomorphic encryption, Onion routing |
| **Unlinkability VS Integrity** | Cryptographic |
| **Pseudonymity VS Confidentiality** | Searchable encryption |
| **Undetectability VS Confidentiality** | Steganographic technologies |

**2- Techniques suitable for Privacy requirements only, or Security requirements:**

| PRIVACY REQUIREMENTS | TOOL TO SUPPORT REQUIRMENT |
|---|---|
| **Anonymity** | Cryptographic, Steganographic technologies, Onion routing, trusted third parties, Dummy traffic, Zero-Knowledge Proofs of Knowledge (ZKPoKs), K-anonymity |
| **Unobservability** | Dummy traffic |
| **Pseudonymity** | Searchable encryption, Public key |
| **Undetectability** | Dummy traffic, Steganographic technologies |
| **Unlinkability** | Cryptographic, Steganographic technologies, Homomorphic encryption, Onion routing, K-anonymity, data hiding, trusted third parties, dummy traffic. |

| SECURITY REQUIREMENTS | TOOL TO SUPPORT REQUIRMENT |
|---|---|
| **Confidentiality** | Cryptographic, accesses control enforcement, Symmetric key and public key encryption, Steganographic technologies, Homomorphic encryption, Onion routing, Searchable encryption |
| **Integrity** | Cryptographic, accesses control enforcement, message authentication codes (MAC) redundancy and comparison |
| **AVAILABILITY** | Redundancy to the system |

## *D. EVALUATE OUTCOME RESULTS:*

Conflict outcomes depend on the method used to deal with the conflict. Each conflict has desired output and stakes in interpersonal relationships, so these factors help determine the issue as well [23].

The first possible outcome is a **Collaboration**, wherein all desired output is achieved. It is a win/win strategy because relationships between both parties are preserved or strengthened while reaching the goal of the project. Another possible outcome is a **Compromise**. This outcome provides partial satisfaction for both parties as a fraction of their desired result is achieved. A Compromise al so preserves the relationships between both parties but can also cause minor strain during negotiations. The third outcome occurs when one party yields to the other: **Accommodation**. The yielding party conforms to the demands of the other party to preserve o r strengthen their relationships. The opposite of this outcome is **controlling**, wherein a party forces the other to follow their desired outcome. This outcome often results in consequences to the losing party, placing a significant strain on the relationship between the parties. Finally, **Avoiding** is an outcome that happens when both parties quit and abandon their

desired results as well as their working relationship. In this case, both parties lose due to the withdrawal.

## V. CONCLUSIONS AND FUTURE WORK

We proposed a framework to identify and mitigate conflicts between security and privacy requirements, being motivated by previous studies that illustrated the problem with a lack of solution. The framework have four sequence steps: The first step of identifying conflicts is to spot an attention to this problem from the literature to realize about conflicting issue. After that we focused on the most frequents requirements being conflicts to set our priority in order to resolve conflicts. In order to resolve conflicts we should follow some strategies that would be one of framework outcome. Therefore first we should give each requirement a value and do a prioritization for requirements in order to decide which requirement the framework should support. After that we present some tools to support requirements base on its natural. Final section we have the evaluation criteria to measure whether the resolution method solve the problem or at least reduce it? Or we have to reconsider using different resolving method. In the future, we will describe more about prioritization techniques and how to set a criteria to choose the appropriate technique. After that we will apply this framework on E-Health case study, to observe the problem and resolution techniques , in order to validate those solution we have to apply it in a real case study in order to validate effectiveness of the framework and whether it needs more improvements.

## REFERENCES


[1] Albrecht, Jan Philipp. "How the GDPR will change the world." *Eur. Data Prot. L. Rev.* 2 (2016): 287

[2] E. Paja, F. Dalpiaz, and P. Giorgini, "Managing security requirements conflicts in socio-technical systems," in International Conference on Conceptual Modeling, 2013, pp. 270-283: Springer.

[3] Q. Ramadan, et al., "Detecting Conflicts Between Data-Minimization and Security Requirements in Business Process Models." *European Conference on Modelling Foundations and Applications.* Springer, Cham, 2018.

[4] Egyed, Alexander, and Barry Boehm. "A comparison study in software requirements negotiation." Proceedings, INCOSE'98. 1998.

[5] Aldekhail, Maysoon, Azzedine Chikh, and Djamal Ziani. "Software Requirements Conflict Identification: Review and Recommendations." International Journal of Advanced Computer Science & Applications 1.7 (2016): 326-335.

[6] Mairiza, Dewi, and Didar Zowghi. "An ontological framework to manage the relative conflicts between security and usability requirements." Managing Requirements Knowledge (MARK), 2010 Third International Workshop on. IEEE, 2010.

[7] Mairiza, Dewi, et al. "Conflict characterization and analysis of non functional requirements: An experimental approach." Intelligent Software Methodologies, Tools and Techniques (SoMeT), 2013 IEEE 12th International Conference on. IEEE, 2013.

[8] Liu, Julie Yu-Chih, et al. "Relationships among interpersonal conflict, requirements uncertainty, and software project performance." International Journal of Project Management 29.5 (2011): 547-556.

[9] D. Alkubaisy, "A framework managing conflicts between security and privacy requirements." *Research Challenges in Information Science (RCIS), 2017 11th International Conference on*. IEEE, 2017.

[10] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, Non-functional requirements in software engineering. Springer Science & Business Media, 2012

[11] V. Bryl, F. Massacci, J. Mylopoulos, and N. Zannone, "Designing security requirements models through planning," in International Conference on Advanced Information Systems Engineering, 2006, pp. 33-47: Springer.

[12] C. Kalloniatis, E. Kavakli, and S. Gritzalis, "Addressing privacy requirements in system design: the PriS method," Requirements Engineering, vol. 13, no. 3, pp. 241-255, 2008.

[13] C. Kalloniatis, E. Kavakli, and S. Gritzalis, "Designing Privacy Aware Information Systems," Software Engineering for Secure Systems: Industrial and Research Perspectives: Industrial and Research Perspectives, p. 212, 2010.

[14] Diamantopoulou, Vasiliki, et al. "Supporting the design of privacy-aware business processes via privacy process patterns." *Research Challenges in Information Science (RCIS), 2017 11th International Conference on*. IEEE, 2017.

[15] Leydesdorff, Loet. "Redundancy in systems which entertain a model of themselves: Interaction information and the self-organization of anticipation." Entropy 12.1 (2010): 63-79.

[16] Dodis, Yevgeniy, et al. "Message authentication, revisited." Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2012.

[17] O. Berthold and H. Langos, "Dummy traffic against long term intersection attacks." In Dingledine, Roger and Syverson, Paul, editors, Proceedings of Privacy Enhancing Technologies workshop (PET 2002). Springer-Verlag, LNCS 248.

[18] SHUANTAE, ANG YU TING. ADDING FUEL TO FIRE: WHEN LAY THIRD-PARTY CONFLICT RESOLUTION STRATEGIES ESCALATE INTERGROUP CONFLICTS. Diss. 2018.

[19] Gentry, Craig, et al. "Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs." *Journal of Cryptology* 28.4 (2015): 820-843.

[20] Röpke, Luise. "The development of renewable energies and supply security: a trade-off analysis." *Energy policy* 61 (2013): 1011-1021.

[21] Deng, Leiwen, and Aleksandar Kuzmanovic. "Pseudonymous public keys based authentication." U.S. Patent Application No. 13/154,125.

[22] Kumar, Arvind, and Km Pooja. "Steganography-A data hiding technique." *International Journal of Computer Applications* 9.7 (2010): 19-23.

[23] N. Katz and K. McNulty, "Conflict Resolution", 1994, pp.