Managing Security Trade-offs in the Internet of Things Using Adaptive Security

Waqas Aman and Einar Snekkenes
Norwegian Information Security Laboratory (NISLab)
Gjøvik University College, Norway
Email: {waqas.aman, einar.snekkenes}@hig.no

***Abstract*—Adaptive security can take dynamic trade-off decisions autonomously at runtime and is considered a key desirable attribute in the Internet of Things (IoT). However, there is no clear evidence that it can handle these trade-offs optimally to add value to such a complex and dynamic network. We present a scenario-based approach to recognize and evaluate typical security trade-off situations in the IoT. Using the Event-driven Adaptive Security (EDAS) model, we provide the assessment of dynamic trade-off decisions in the IoT. We have showed that an optimum trade-off mitigation response in the IoT can be automated by assessing various contextual requirements, such as the QoS and user preferences, thing capabilities, and the risk faced, at runtime. eHealth scenarios are examined to illustrate system application in IoT-based remote patient monitoring systems.**

Keywords— Internet of Things; Adaptive Security; eHealth; Event Driven Architecture.

I. Introduction

IoT has a huge potential to facilitate the growth of our economy and society by digitizing commercial enterprises and public infrastructures. The European Commission envisions the market value of IoT to be one trillion euros by the year 2020 [1], yet alone in the Europe. IoT aims to connect diverse technologies, objects, services and people to achieve particular objectives. This interconnection introduces heterogeneity, complexity and dynamic elements in the concerning service architecture.

From a security perspective, these heterogeneous things in the IoT ecosystem have their inherited vulnerabilities and connecting them together will open a multitude of new means and opportunities for the adversaries. Hence, this diversity makes the IoT threat landscape more complex though provides flexibility. Such a broad threat spectrum may not be addressed by the conventional security controls as they are designed to protect against a particular threat context, such as particular files or network packets. Their risk mitigation strategies are primarily focused on asset protection and do not consider other factors, such as resource capacity, QoS requirements, and user preferences, which are critical for a user-centric IoT-based service. The resulting decisions can be inflexible and inefficient and may negatively influence the monitored service. Furthermore, due to the increasing number of objects per user in the IoT [2], it will be relatively difficult to implement manual risk management activities.

The mentioned problems motivate autonomic security adaptation, a key desirable attribute in IoT-enabled smart environments [3]. In the IoT, adaptive security can be employed to achieve a cost-effective trade-off decision to reduce risks faced at runtime. Such attempts will significantly improve the overall service reliability as it would appraise all the potential factors affecting or affected by the decision. However, due to the IoT architectural complexity, it is challenging to recognize, assess and model potential trade-off situations using adaptive security. To address adaptive security in IoT, we have proposed and analyzed the feasibility of Event-driven Adaptive Security (EDAS) architecture in [4] and [5]. In this article, we explicate a scenario-based method to evaluate various security tradeoffs using EDAS. Our emphasis is to investigate two essential questions: i) What typical trade-off situations exist in the IoT? And, ii) To what extent does the EDAS adaptive security loop add value to autonomic risk management in the IoT?

We have found that by using EDAS, security adaptation in IoT can be effectively automated by utilizing a scenario-based approach. The mitigation response it adapts examines all the potential contextual requirements, i.e. QoS requirements, user preferences, resource capacity, and threat level. Hence, the response it adapts reflect an optimum trade-off decision as it weighs all the influencing factors and selects the one which has a maximum utility. Furthermore, the approach used in this article will empower system analysts and developers to identify and evaluate key pre-development requirements, e.g. context awareness essentials, trade-off metrics, and conflicts, programming aspects, etc., that are critical for engineering event-driven adaptive security. Moreover, it is realized that a more precise set of trade-off metrics need to be developed and analyzed to capture the contextual requirements accurately and for the adaptation decision to be more efficient. IoT-enable eHealth scenarios are investigated to reflect EDAS application.

The rest of the paper is organized as follows. A brief introduction to EDAS and the approach used in this paper is given in Section II. The IoT-eHealth scenarios and corresponding trade-offs are briefly described in Section III. Section IV details a schema of how the scenarios and trade-offs can be modeled in the EDAS. In Section V, we will discuss some of the adaptation concerns and will relate them to work done in the literature. Finally, the article is concluded in Section VI.

II. Architecture and Approach

This section briefly introduces the EDAS model and describes the approach used to recognize and assess the potential trade-offs using EDAS.

A. The EDAS Model

An Event-driven Architecture (EDA) collects, analyzes and reacts to significant changes, events, in the monitored network. Monitoring these events provide a holistic visibility of the operations across the network. The primary feature offered by an EDA is loose-coupling which enables the system components to operate independently [6]. Hence, it offers flexibility, interoperability and extensibility in the design, which are highly desirable attributes in IoT-related architectures. The Event-driven Adaptive Security (EDAS) is an autonomic security adaptation model based on EDA [4]. Its reference model is depicted in Fig. 1. EDAS monitors, analyzes and responds to security threats (*thing*-generated events) using a continuous control feedback loop [7]. The *Risk Monitor* component collects, filters and normalizes events, before or after adaptation, emerging from the monitored *Event Sources* (*things*) in the IoT. The *Risk Analyzer* investigates different events in a context by correlating them for possible threats and raises a risk alarm when a threat discovered has a risk level beyond the threshold. The *Risk Adapter* utilizes a runtime adaptation ontology and responds to an alarm by selecting an optimal response as per the contextual requirements. The model is extended to a technical specification of a system architecture, and its feasibility is investigated as a real-world artifact in [5]. A description of its major components is given in Table I. The relationships among these components is detailed in Section IV.

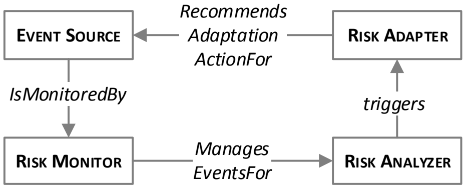


Figure 1: EDAS Reference Model

B. Towards Adaptive Security

We consider an adverse security scenario as a trade-off situation as there is always *security vs. some attribute* trade-off involved when mitigation actions are adapted to reduce the risk faced. The two-phased approach used to elicit and analyze engineering fundamentals of EDAS is depicted in Fig. 2 and is briefly described as follows:

The *Phase-1* focuses on the knowledge elicitation and evaluation required to identify, assess, and respond to potential threats in the corresponding scenarios. This knowledge includes threats, critical event sources, event correlation contexts, participating events, risk analysis methods, supported adaptable actions, trade-off metrics, conflicting scenarios and their resolution approaches against the individual scenarios.

The *Phase-2*, scenario modeling, in this article, refers to the pre-development realization of the knowledge gathered in the Phase-1. It is performed by populating the adaptive security system model with the knowledge extracted. The realization can serve as an implementation guideline for the analysts and developers, and assist them in identifying and evaluating different development paradigms for each scenario.

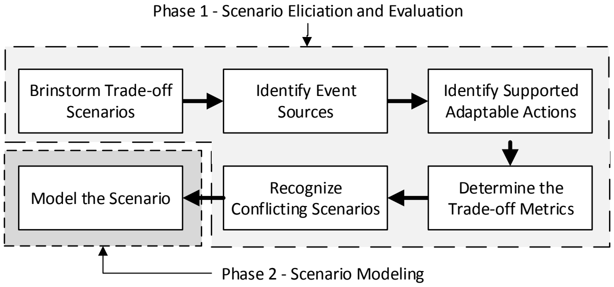

Phase 1 - Scenario Eliciation and Evaluation
Brinstorm Trade-off Scenarios
Identify Event Sources
Identify Supported Adaptable Actions
Determine the Trade-off Metrics
Recognize Conflicting Scenarios
Model the Scenario
Phase 2 - Scenario Modeling

Figure 2: A Scenario-based Approach Towards Adaptive Security

III. SCENARIOS AND ADAPTATION TRADE-OFFS

In this section, we instantiate the Phase-1 of the approach with a few typical IoT-eHealth scenarios to highlight the trade-offs and to reflect on the overall process. We extend the IoT-eHealth case study in [4] and add different scenarios to narrate various real-world security incidents.

A. The IoT-eHealth Case Study and Scenarios

A hypoxemic patient at home, Lynda, equipped with an Oximeter is monitored from a remote hospital site. She has a smart device capable of mobile and internet-based communication. It has some general purpose sensors, such as a GPS sensor, and is used in activities like conferencing with the physicians, viewing health stats and prescriptions, billing and payments. Moreover, it acts as a relay access point between the sensors and the hospital and ensures that vital body signs are available during outdoor activities.

Scenario 1 – Resource optimization during mobility: Before going outdoors for a prescribed exercise, Lynda changes the smartphone settings from WiFi to Mobile-Data indicating a change in operational context. As increased encryption consumes more power and memory, confidentiality has to be reduced as per the utility to ensure long-term data availability

Scenario 2 - Max. Confidentiality in Possible Intrusions: Assuming discovering unregistered radio devices as a threat to confidentiality, the patient requirements and the hospital policy dictate that confidentiality has to be increased in to avoid any possible compromise. This scenario is identified as *2a* and *2b* in the home and outdoor operational contexts respectively.

Scenario 3 - Handling a *thing* Compromise: The network component of the eHealth app on the patient smart device has somehow been compromised. The app has generated events indicating that a new destination has been added to the address list.

Scenario 4 - Repeated Wrong Login Attempts: An adversary having physical access to Lynda's smartphone is trying

Table I. A Description of EDAS Components

	Entity	Description
Event Source	Thing	A physical asset in the monitored IoT ecosystem
	Object	a software module of a *Thing* e.g. a temperature sensing module
	Security Component	Security Mechanism e.g. algorithms, used by an *Object*
	Event	A potential change in the *Thing* environment raised by an *Object*
	Event Framework	The *Event* handler and logger
	Local Adopter	A software module that instructs the execution the adaptation decision locally
	Adapt Request	The adaptation decision/action (risk mitigation response) to be adopted locally
Monitor	Monitoring Agent	A software component that collect, filter and transform events
	Filtration Criteria	An event filtration rules
	Normalization Criteria	Event transformation rules
Analyzer	Alarm	Risk alert detailing risk beyond acceptance
	Risk Scorer	Event risk quantifier and *Alarm* generator
	Risk Metric	A measure based on which risk is quantified, e.g. an asset or event importance value
	Threat Context	A marker specifying a particular risk situation
	Correlation Directive	A container for a rule set that directs risk manipulation for a *Threat Context*
	Correlation Criteria	Rules that correlate events in time and space
Adapter	Action	A possible risk mitigation response
	Mechanism	A vocabulary of the monitored ecosystem's security method, e.g. routing or encryption algorithms
	Property	A vocabulary of the attributes inherited by the *Mechanism*, e.g. key length
	Utility Metric	A trade-off factor influencing or influenced by a property to be adopted
	Utility	A positive integer indicating the extent to which a Metric is supported
	Risk Level	Risk impact level
	Contextual Requirement	Preferences and capabilities in a particular operational environment/context

random passwords to login into the eHealth app installed to steal the banking information stored in it.

Scenario 5 - Physician Account Compromise: A Physician has successfully logged on to the Electronic Health Record (EHR) server from his machine. However, no such record is found in the employee attendance (RFID) server. Besides a technical fault, the situation indicates that the account might have been compromised.

Scenario 6 - Service Unavailability: The EHR server at the hospital, the primary destination for the remotely collected vital signs, suddenly goes down due to a technical fault. In such situations, the smart device has to store vital sign information locally.

Table II depicts the organization of the adaptation knowledge obtained in Phase-1. Fig. 3 shows a general view of the primary trade-offs involved in each scenario with the possible adaptation actions having distinct utilities in a trade-off as per the contextual requirements. In EDAS, an adaptable action comprises a *security mechanism* and its *property*, such as the AES encryption algorithm and its 128-bit key length property, supported by a particular event source. At a given time in a particular operational context, a property addresses a particular risk level. Metrics influenced in a trade-off, as shown in Table II, are derived from the contextual requirements and are weighed against each property to reflect its overall utility and are different in different operational contexts. All these elements will be further explored in the next section to reflect on how they are addressed in EDAS.

As an example, in Table II, we have identified two conflicting scenarios (1 and 2b) as both will compete for the conflicting requirements, i.e. availability and confidentiality, in outdoor situations. Some conflicts may not be critical and can easily be resolved by simple *if-else* related techniques. For instance, one can ignore scenario 2b if scenario 1 has already occurred as it has, comparatively, more importance for service and user. However, other conflicts might need in-depth investigations requiring more sophisticated resolution mechanisms.

IV. Scenario Modeling

Scenario modeling serves two primary purposes. First, it provides a platform for the analysts to realize the knowledge evaluated in Phase-1 and assists in identifying any missing information. Thus, it further evaluates the adaptation knowledge required to analyze a threat scenario. Secondly, it will provide a guideline for the developers to better understand problem (scenario) requirements for implementation and will facilitate them to identify and evaluate different programming techniques.

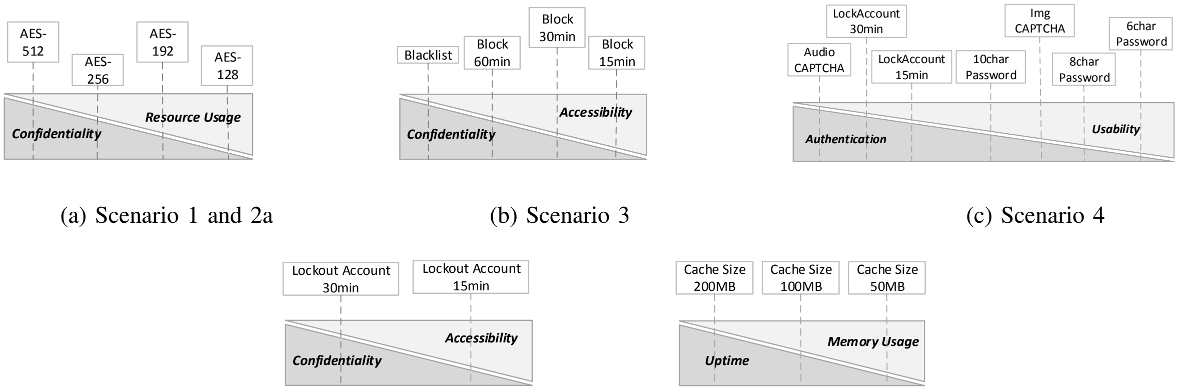
Taking scenario 1 and 2b as examples, we illustrate the scenario modeling and reflect on how the corresponding knowledge relates to each other. We present two illustration views: Fig. 4 depicts a tabular description of the concerning relations in Event Source whereas, Fig. 5-7 provides a conceptual view of the corresponding components. These figures extend the reference model (Fig. 1), provide a blueprint of the relationship between the major components, and describe how the extracted knowledge in Phase-1 can be structured for EDAS implementation.

The Event Source represents the monitored resource in the ecosystem. It consists of a physical asset (a *thing* in the IoT), and application specific objects. These objects generate events using their event framework facility and send them to the remote EDAS platform for threat analysis. The platform includes the risk monitor, analyzer and adapter components. An object does not take the adaptation decision by itself, but receives it as a request from the adapter via the local adopter and implements it locally. Using the scenarios knowledge, the relation between the Event Source components is shown in Fig. 4.

In Fig. 5, the strings starting from *Acpt* can be considered as regular expressions (RegEx) or rules to be designed to accept a

Table II. Scenario Elicitation and Evaluation

Sc. No.	Opr. Context	Associated Threat	Possible Event Sources	Supported Adaptable Actions	Supported Adaptable Mechanism[Properties]	Trade-off Metrics	Conflict
1	Outdoor	Data Unavailability	Oximeter, Smart phone	Change Cipher Change Cipher Key Length	Cipher[AES] Keylength [128, 192, 256, 512]	Efficiency, Resource Usage, Confidentiality	2b
2a	Home	Privacy & Confidentiality Breach	Oximeter, Dev Detector Sensor	Change Cipher Change Cipher Key Length	Cipher[AES] Keylength [128, 192, 256, 512]	Efficiency, Resource Usage, Confidentiality	
2b	Outdoor						1
3	Indoor, Outdoor Hospital	Information Hijacking	Smart phone, Management Server	Block ID/Address	Permanent[blacklist], Temporary[15min, 30min, 60min]	Accessibility, Confidentiality	
4	Indoor, Outdoor Hospital	Password Guess/Brute force Attack	Smart phone	Change Password Length, Lock Account, Enforce CAPTCHA	Length[8char, 10char], Lock Time[15min, 30min], CAPTCHA[Audio, Image]	Memorability, EaseOfUse, Accessibility, Authentication, Resource Usage	
5	Hospital	Intrusion	RFID Server, EHR Server	Change Account settings	LockAccount[15min, 30min],	Accessibility, Authentication	
6	Home, Outdoor, Hospital	Service Unavailability	Smart phone, EHR Server	Activate Local Cache	Cache Size[50MB, 100MB, 200MB]	Distress, Memory, Uptime, Energy usage	



(a) Scenario 1 and 2a

(b) Scenario 3

(c) Scenario 4

(d) Scenario 5

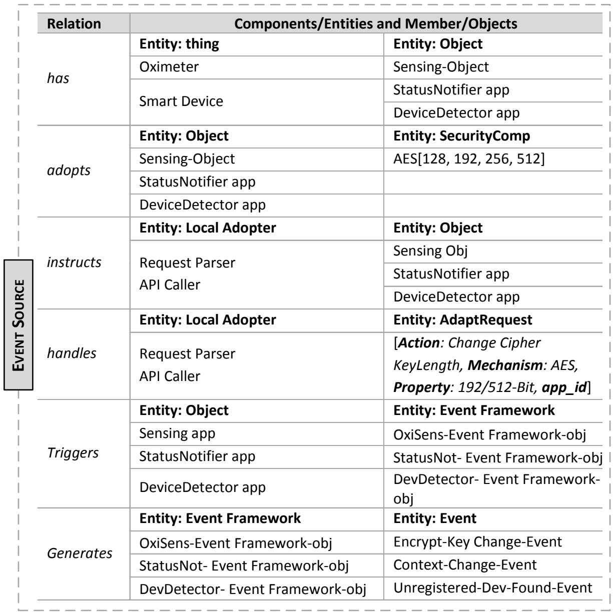
(e) Scenario 6

Figure 3: Scenarios, Primary Trade-offs, Adaptation actions & their utilities

particular event for further analysis. Normalization rules apply specific transformation rules to each event, depending upon its origin and importance, for further analysis. These strings and tags in the modeling provide a precise instruction set for the developers to construct the essential components. Therefore, this schematic modeling reduces communication gap between system analysts, architects, and developers, and speeds up the engineering process.

Each normalized event from the Monitor has associated risk metrics based on which the *Quantifier* object in the Risk Analyzer, see Fig. 6, calculates its risk. These metrics may also be modified during event correlation. Event correlation can also be used to investigate and resolve any conflicting scenarios. For instance, the *Correlation Criteria*, in Fig. 6, resolves the conflict between Scenario 1 and 2b by correlating the operational context. Moreover, it can be noticed that *Encrypt-Key-Change-Event* is also participating in the correlation contexts. Depending on the context, it represents the event that has been raised by the Oximeter sensing object after new encryption key lengths are adapted and is correlated in the same threat context to ensure that the threat has been addressed, and that the corresponding risk level has been reduced as per the contextual requirements. The `INCREASE` and `NORMALIZE` keywords specify the particular function calls or related equations that can be employed to manipulate the risk level as per the acceptance threshold. Furthermore, as event correlation intends to analyze events from different sources, it may include other sources which might not be a direct target in the threat faced but may provide essential information for correlation. Thus, the correlation criteria modeling enables the analysts to discover and assess other sources that may be critical in analyzing scenarios.

The Risk Adapter components, as shown in Fig. 7 (excluding the *Object* and *Alarm*), are the necessary vocabulary in



EVENT SOURCE

Relation	Components/Entities and Member/Objects	
has	**Entity: thing**	**Entity: Object**
	Oximeter	Sensing-Object
	Smart Device	StatusNotifier app
		DeviceDetector app
adopts	**Entity: Object**	**Entity: SecurityComp**
	Sensing-Object	AES[128, 192, 256, 512]
	StatusNotifier app	
	DeviceDetector app	
instructs	**Entity: Local Adopter**	**Entity: Object**
	Request Parser API Caller	Sensing Obj
		StatusNotifier app
		DeviceDetector app
handles	**Entity: Local Adopter**	**Entity: AdaptRequest**
	Request Parser API Caller	[***Action***: *Change Cipher KeyLength*, ***Mechanism***: *AES*, ***Property***: *192/512-Bit*, ***app_id***]
Triggers	**Entity: Object**	**Entity: Event Framework**
	Sensing app	OxiSens-Event Framework-obj
	StatusNotifier app	StatusNot- Event Framework-obj
	DeviceDetector app	DevDetector- Event Framework-obj
Generates	**Entity: Event Framework**	**Entity: Event**
	OxiSens-Event Framework-obj	Encrypt-Key Change-Event
	StatusNot-Event Framework-obj	Context-Change-Event
	DevDetector- Event Framework-obj	Unregistered-Dev-Found-Event

Figure 4: Event Source (tabular view)

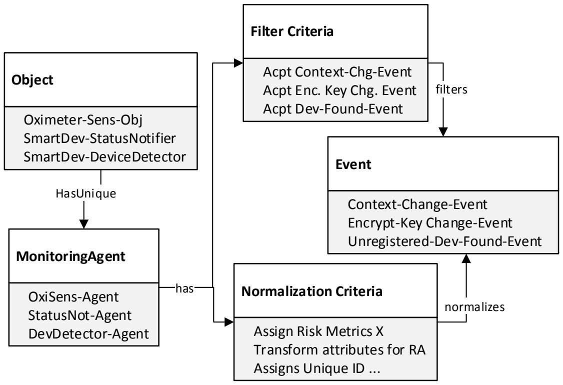


Figure 5: Risk Monitor (conceptual view)

the adaptation ontology proposed in [5]. It is accessed as per the scenario to formulate the adaptation (trade-off) decision which is sent to the Event Source as an adaptation request (*AdaptRequest*).

A. Managing Trade-offs

Taking decisions always involves one or more trade-offs. The corresponding influences can sometimes be very low and can be ignored. For instance, while weighing various security metrics for an adaptation action to appropriately control access, e.g. changing a password length to 10 characters, the data integrity or confidentiality metrics can be disregarded as it has no significant influence on the decision. However, there will be situations that will require careful assessment of the influencing parameters to address all the potential requirements appropriately.

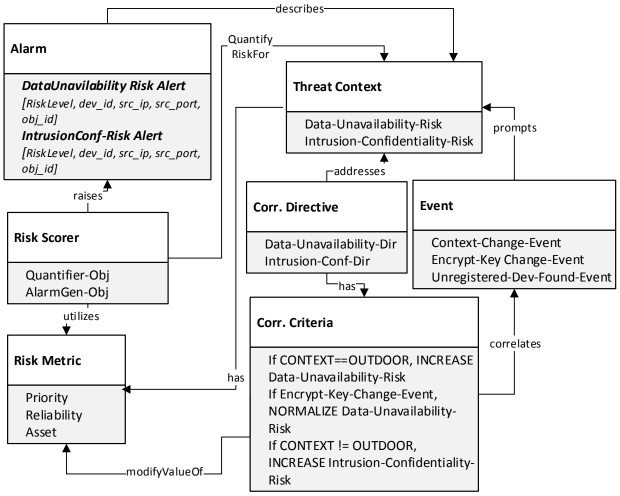

describes
Alarm
DataUnavailability Risk Alert
[RiskLevel, dev_id, src_ip, src_port, obj_id]
IntrusionConf-Risk Alert
[RiskLevel, dev_id, src_ip, src_port, obj_id]
Quantify RiskFor
Threat Context
Data-Unavailability-Risk
Intrusion-Confidentiality-Risk
prompts
raises
addresses
Risk Scorer
Quantifier-Obj
AlarmGen-Obj
Corr. Directive
Data-Unavailability-Dir
Intrusion-Conf-Dir
Event
Context-Change-Event
Encrypt-Key Change-Event
Unregistered-Dev-Found-Event
has
utilizes
Corr. Criteria
Risk Metric
Priority
Reliability
Asset
has
If CONTEXT==OUTDOOR, INCREASE Data-Unavailability-Risk
If Encrypt-Key-Change-Event, NORMALIZE Data-Unavailability-Risk
If CONTEXT != OUTDOOR, INCREASE Intrusion-Confidentiality-Risk
correlates
modifyValueOf

Figure 6: Risk Analyzer (conceptual view)

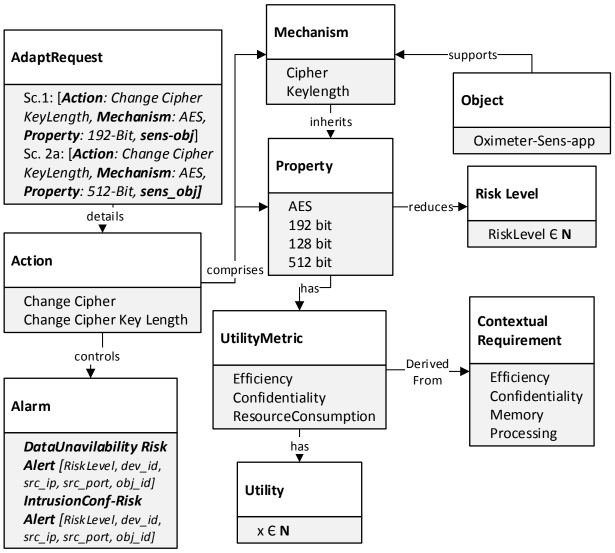


Figure 7: Risk Adapter (conceptual view)

In EDAS, factors involved in a trade-off are considered as utility metrics, as shown in Fig. 7. They are derived from the contextual requirements identified in the monitored IoT ecosystem, i.e. user preferences, QoS requirements, and *thing* resources, and can have different utilities in different operational contexts. For instance, confidentiality, integrity, and availability requirements may differ significantly in outdoor contexts because of the adverse elements in the environment as compared to the home context. However, the usability requirements might remain the same in almost all contexts. For each property used in an action, these metrics are assessed, i.e. assigned a utility (a positive integer) by experts based on the property's competence against the threat and its influence on the contextual requirements. The greater the integer value is, higher is the utility of a metric. This assessment facilitates the system, i.e. the Risk Adapter in EDAS, to take a trade-off decision that has a maximum utility in a particular threat scenario and is, thus, an optimum response.

As examples, depicted in Table III-V, we illustrate how some trade-offs concerning Scenario 1, 2a, 4 and 6 can be handled in a security adaptation decisions. It can be noticed that we have expanded the primary trade-offs (in Fig. 3) to influencing metrics at the abstract level in each scenario to address the possible contextual requirements. However, in practice, these metrics should reflect all the influencing and influenced contextual pre-requisites for the decision to be more effective. The property with the highest total utility is selected, shaded out in gray, as the most cost-effective mitigation action to confront a threat in a scenario.

V. Discussion and Related Work

In this section, we discuss a few concerns, such as the trade-off metrics assessment and design restrictions, and relate them to similar work in the literature to comprehend how the concepts proposed in the EDAS or related work can benefit from each other to make adaptive security a more reliable solution for the IoT.

A. Trade-off Metrics Assessment

At this stage of EDAS development, we have not investigated any particular trade-off metrics. However, our scenario-based approach suggests how they can be recognized in the IoT. We emphasize that all contextual requirements should be identified in potential operational contexts and should be categorized rigorously to capture the actual needs. A rigor classification of the requirements will result in a precise set of trade-off metrics and will make the adaptation decision more realistic and, therefore, effective. For instance, a patient usability preference should be further extended to other factors, such as learnability, memorability, ease of use, satisfaction, etc., to carefully address his preferences in concerning scenarios.

The metrics assessment method during adaptation decision is also critical. Since the primary objective of EDAS was to provide a holistic autonomous security architecture, we did not investigate the effectiveness of its utility-based metric assessment. Although, it does offer a rationale for optimized adaptation decision, we have yet to explore it further for any improvements. In this context, methods from game theory [8], expected utility theories, machine learning, and related studies may provide significant and useful perspectives.

Depending on the organizational policy, the selection of a property can be approached in two ways. If the total utility of two or more properties has the same value, one of them can be randomly adapted as it implies that they all have the same maximum utility in a given context. Otherwise, conflicts may arise due to utility overlapping which will necessitate more sophisticated assessment methods as mentioned earlier. Therefore, more meaningful and structured values (utilities) should be established to weigh individual metrics. In this regard, methods defined in [9], [10] and [11] can be potentially reviewed for developing and estimating metrics.

B. The Evaluation Approach

Similar evaluation frameworks can be found in the literature assessing different security and privacy aspects in information systems. The Architecture Tradeoff Analysis Method (ATAM) [12] suggested a scenario-based approach to analyze design approaches addressing various QoS attributes in software architectures. A similar approach is used in [13] where the authors utilized a scenarios-based method to evaluate the security of a software architecture. Recently, a more relevant evaluation framework is suggested by Liester et.al. [14]. The authors have provided an extensive list of IoT-eHealth scenarios as various system states. Linear and logarithmic approaches were utilized to assess and quantify their security and QoS requirements in an adaptive security system. Our approach complements their work and emphasizes to actively consider user preferences and devices capabilities besides QoS and security requirements to make the adaptation decision more effective. Furthermore, our approach tends to model the requirements in a way such that they can be easily and readily employed in the system development and implementation.

C. Architectural Constraints

From an architectural viewpoint, not every object is adaptable. In EDAS, only those objects can be adapted which utilize a flexible security component. Although, some objects are critical to security, they are only used to collect essential events for establishing context-aware analysis, e.g. a GPS module. Such objects may not use any security component. Others may have only a single supported security component, e.g. a DES-128 bits encryption algorithm. Apparently, in such cases, security adaptation does not seem to be practical. However, a possible trade-off in such scenarios can be that of a *zero encryption level* indicating an adaptation decision that instructs to drop any security mechanism in use. Evidently, this is not an efficient protection strategy, but can be useful in situations where confidentiality is not the primary objective, e.g. outdoor emergency scenarios where the patient's data availability is more critical than its confidentiality. To ensure flexible and more optimized adaptation, other design elements, such as the sensor middleware in the Global Sensor Network (GSN) [15] and related middlewares, could be introduced into the architecture. Such middlewares can be used to offer flexible security components as services for objects having none or reduced security components.

VI. Conclusion

Adaptive security is a desirable attribute in the IoT where the threat landscape is more complex and dynamic. In this paper, we have provided a scenario-based method that will facilitate system architects, analysts and developers to identify and evaluate different aspects of engineering event-driven adaptive security in the IoT. Using event-driven adaptive

Table III. Trade-off Assessment - Scenario 1 and 2a (Security = Confidentiality). Assuming 256-bit key is used before adaptation

Tradeoff/Utility Metric	***Scenario 2a*** ***Context = Home/Hospital***				***Scenario 1*** ***Context = Outdoor***			
	Mechanism =AES-Key Length							
	Properties							
	128-bits	*192-bits*	*256-bits*	*512-bits*	*128-bits*	*192-bits*	*256-bits*	*512-bits*
Security	10	15	18	21	10	15	18	21
Efficiency	15	14	13	12	15	14	13	12
Resource Usage	17	16	15	14	17	14	10	6
Total Utility	***42***	***45***	***46***	***47***	***42***	***43***	***41***	***39***

Table IV. Trade-off Assessment - Scenario 4 (Security = Authentication)

Trade-off/Utility Metric	***Mechanisms***					
	Key Length		**CAPTCHA**		**Time Restriction**	
	Properties					
	8-Char	*10-Char*	*Image*	*Audio*	*15min*	*30min*
EaseOfUse	10	8	20	18	10	5
Memorability	15	10	2	2	2	2
Accessibility	10	7	20	10	10	5
Security	10	15	10	12	10	15
Resource Usage	12	12	8	5	12	12
Total Utility	***57***	***52***	***60***	***47***	***44***	***39***

Table V. Trade-off Assessment - Scenario 6 (Uptime = Security)

Trade-off/Utility Metric	*Context = Home/Hospital*			*Context = Outdoor*		
	Mechanisms = Cache Size					
	Properties					
	50MB	*100MB*	*200MB*	*50 MB*	*100MB*	*200MB*
Distress	20	10	5	15	8	4
Uptime	15	25	30	10	15	20
Memory	20	10	5	20	15	10
Energy Usage	10	10	10	15	10	5
Total Utility	***65***	***55***	***50***	***60***	***48***	***39***

security (EDAS), and a few typical IoT-eHealth scenarios, we have provided essential knowledge that optimal trade-off adaptation decisions can be employed in the IoT to defend against a risk faced. Therefore, it is made evident that adaptive security can improve autonomous risk management in the IoT by adequately addressing the trade-offs. We have utilized a utility-based assessment method to deal with the trade-off metrics involved in a decision. Since these metrics are derived from the monitored ecosystem requirements, an adaptation decision evaluating these requirements results in a trade-off decision which is the most effective in a given scenario. The assessment provides a convincing basis for making dynamic trade-off decisions.

Acknowledgment

This work is supported by the Adaptive Security for Smart Internet of Things in eHealth(ASSET, 2012/15) project funded by the Norwegian Research Council under Grant Agreement no. 213131/O70. We wish to thank our colleagues and the anonymous reviewers for their expert opinion.

References

[1] IDC Italia S.r.L and TXT e-solutions S.P.A., "Definition of a research and innovation policy leveraging cloud computing and iot combination," European Commission DG Communications Networks, Content & Technology, Tech. Rep., May 2015.

[2] D. Evans, "The internet of things:how the next evolution of the internet is changing everything," CISCO, Tech. Rep., April 2011, last accessed on: 31 Aug 2015. [Online]. Available: http://bit.ly/1Inzh2Q

[3] H.-D. Ma, "Internet of things: Objectives and scientific challenges," *Journal of Computer science and Technology*, vol. 26, no. 6, pp. 919–924, 2011.

[4] W. Aman and E. Snekkenes, "Event driven adaptive security in internet of things," in *UBICOMM 2014, The Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2014, pp. 7–15.

[5] W. Aman and E. Snekkenes, "EDAS: An evaluation prototype for autonomic event-driven adaptive security in the internet of things," *Future Internet*, vol. 7, no. 3, pp. 225–256, 2015.

[6] B. M. Michelson, "Event-driven architecture overview," *Patricia Seybold Group*, vol. 2, 2006. [Online]. Available: http://bit.ly/1hadIqX

[7] W. R. Ashby, *An introduction to cybernetics*. Chapman & Hall Ltd., 1956.

[8] R. B. Myerson, *Game theory*. Harvard university press, 2013.

[9] B. G. Marcot, "Metrics for evaluating performance and uncertainty of bayesian network models," *Ecological modelling*, vol. 230, pp. 50–62, 2012.

[10] R. Savola and H. Abie, "Development of measurable security for a distributed messaging system," *International Journal on Advances in Security*, vol. 2, no. 4, pp. 358–380, 2009.

[11] B. Ksiezopolski, "Qop-ml: Quality of protection modelling language for cryptographic protocols," *Computers & Security*, vol. 31, no. 4, pp. 569–596, 2012.

[12] R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lipson, and J. Carriere, "The architecture tradeoff analysis method," in *Fourth IEEE International Conference on Engineering of Complex Computer Systems. ICECCS'98.*, 1998, pp. 68–78.

[13] A. Alkussayer and W. H. Allen, "A scenario-based framework for the security evaluation of software architecture," in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, vol. 5. IEEE, 2010, pp. 687–695.

[14] W. Leister, M. Hamdi, H. Abie, S. Poslad, and A. Torjusen, "An evaluation framework for adaptive security for the iot in ehealth," *International Journal On Advances in Security*, vol. 7, no. 3 and 4, pp. 93–109, 2014.

[15] K. Aberer, M. Hauswirth, and A. Salehi, "A middleware for fast and flexible sensor network deployment," in *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 2006, pp. 1199–1202.