On the Need for Extending MARTE with Security Concepts

Mehrdad Saadatmand*, Antonio Cicchetti*, Mikael Sjödin*
*Mälardalen Real-Time Research Centre (MRTC)
Mälardalen University, Västerås, Sweden
{mehrdad.saadatmand, antonio.cicchetti, mikael.sjodin}@mdh.se

***Abstract*—Security has often been considered as an added feature to the design of systems and this has been the root of many security issues. The need for introducing and considering security along with other aspects of the system from early design phases has now gained much more attention and been the subject of many studies. In model-driven engineering, this has also led to the introduction of several modeling languages for security. However, in embedded systems domain due to the limitations of resources, security requirements and features cannot be considered in separation as they impact other aspects such as schedulability, performance and power consumption. MARTE as a modeling language for real-time embedded systems contains the concepts for modeling these aspects, and hardware and software entities. In this paper, we discuss potentials and suitability of MARTE for extension with security concepts and how it can serve as a common framework for modeling security in embedded systems and performing trade-off analysis with other aspects.**

I. Introduction

Considering security as addition of features to a system, not only can lead to inefficient and not optimal use and integration of security mechanisms, but can also impair the design and quality of the system. This issue is more critical in embedded systems that require careful balance among different properties due to resource constraints. Therefore, security should be considered from early phases of design and as a new dimension and metric [1]–[3]. In model-driven engineering which can help with raising the abstraction level and cope with the design complexity of embedded systems, several modeling languages for security have been defined to bring security concepts into the design models of the system. However, due to the characteristics of embedded systems, as will be discussed, security requirements cannot be considered in separation from other requirements. In order to achieve this, modeling language(s) used to design the system should be able to cover these variety of requirements.

MARTE [4] has set a firm ground in modeling non-functional requirements in real-time embedded systems. Considering that MARTE provides concepts for modeling generic non-functional requirements and also has dedicated (sub)profiles for modeling timing, software and hardware resources, we believe security deserves special attention in MARTE for the following reasons:

- The awakeness and shift towards considering and bringing security aspects in higher levels of abstraction and system design.
- Security in embedded systems is of great importance. While security problems in desktop applications could lead to problems such as personal information leaks and financial issues (e.g. credit card, bank systems), in embedded systems they could cause more serious issues such as injuries, death, enemy threats and so on; e.g. in medical systems, military equipments, power plant controllers, public water monitoring systems.
- Security aspects in embedded systems should be modeled along with other non-functional requirements as they incur big impacts and require balance and trade-off analysis with other aspects such as performance, resource usage, availability and schedulability.
- MARTE already has concepts that can be used as basis for definition of security features.

In summary, the appearance of different UML profiles for modeling security in recent years, ever-increasing importance of security in embedded systems, and potentials of MARTE for modeling security (e.g. using concepts in Non-Functional Properties (NFP), Generic Resource Modeling (GRM), Generic Quantitative Analysis Modeling (GQAM), Hardware Resource Modeling (HRM)) all put MARTE in a good position to move towards supporting security modeling. By extending MARTE with security concepts, it becomes possible to have a more precise system model and thus even in case of schedulability and performance analysis which MARTE explicitly supports, a more accurate analysis will be possible (i.e. taking into account impact of security measures).

In this paper, we investigate the benefits of extending MARTE with security aspects and why MARTE is a good choice to build security concepts upon. In section 2, importance and complications of security requirements in embedded system are discussed. A brief look on security modeling is also offered and potentials of MARTE to incorporate security aspects are highlighted. Section 3 pictures an example of an embedded system whose security requirements directly impact other requirements and properties of the system. In section 4 and 5 we focus on encryption and authentication requirements for security in this system and describe our suggested approach to model these features based on MARTE and how it facilitates trade-off and sensitivity analysis. Section 6 focuses on the issue of incorporating security extensions in MARTE and its challenges. Finally in section 7, we give a summary of the

issues covered in the paper and explain future works.

II. Motivation and Potentials

A. Security in Embedded Systems

There are additional challenges in designing secure systems when it comes to embedded systems domain. Embedded systems are supposed to operate as part of other systems (e.g. in vehicles, medical devices, etc.) and they can be used in hostile environments which make them prone to a variety of physical security threats. Also, the big increase in the development of connected embedded devices and their operation in distributed networks require new security considerations during their design [3]. Besides security requirements originating from the usage and operation environment of embedded systems, [1] also discusses other security design challenges unique to embedded systems: security processing gap, assurance gap, battery gap and [security mechanisms] flexibility issue. By looking at these unique challenges from a higher level, we can realize that these requirements mostly originate from the nature of embedded systems which is being constrained in resources such as limitations on power consumption, processing capacity, maintainability, layout and physical dimensions. This leads us to the issue of trade-off between different non-functional properties of embedded systems. For instance, the flexibility issue mentioned in [1] is actually the trade-off between security and maintainability and upgrade-ability of the system. Similarly, battery gap is the trade-off between security mechanisms and available power resources.

B. Security Modeling

There are several efforts on defining UML profiles for security. For example, SecureUML [5] focuses on modeling role-based access control. AuthUML [6] provides a framework for analysis of access control requirements. [7] introduces a set of stereotypes for specification of vulnerabilities that serve as guidelines for developers to avoid them during implementation. UMLsec [8] offers a broader range of security concepts and comes with an analysis tool. [9] tries to offer a solution for modeling security along with timing characteristics of the system using UMLsec and MARTE.

One issue with such profiles and their usage is that most of them are limited in the sense that they usually focus on a certain aspect of security [10]. Therefore in modeling embedded systems especially when they are distributed, we need to apply several different security profiles to cover aspects like authentication, key-exchange, encryption, decryption and access control. This can sometimes be tricky considering that these profiles can have overlapping and conflicting semantics and notations. Also modeling of security requirements is often considered in separation from other requirements such as timing [9], [10]. Besides the ongoing efforts on modeling security, impact analysis and relation of security to other properties of the system is also a challenge.

Another issue with definition and use of separated security profiles is that it is not always straightforward to use a combination of profiles to cover different design aspects of a system; for example, using MARTE and SysML plus SecureUML. Specification and semantic conflicts can occur when combining different profiles [11].

C. Hardware Security

The environments that embedded devices are used in, make them more prone to security issues and types of attacks that are less relevant for other systems. While software security measures for a central database, for example in a company or university, are obviously a necessity, the situation for embedded devices is more complicated. For embedded devices such as flash drives and also mobile phones that can be carried around and contain huge amount of sensitive information from private user data (e.g. billing information for mobile Apps) to manufacturer firmwares, operating systems, and confidential algorithms and codes, security measures more than just software level are required. Therefore, in the design of embedded systems, security measures for physical and side channel attacks (including timing analysis, power monitoring, fault induction and electromagnetic analysis) and requirements on tamper-resistant hardware should also be considered [1], [2].

Considering these issues and also resource limitations in embedded systems, dedicated hardware for security control and also hardware parts with built-in security support are attracting more attention than before. There are manufacturers that design cryptographic hardware accelerators and custom CPUs for lower power consumption and high-performance devices [1], [2], [12]. Advantages and disadvantages of having hardware intrinsic security and awareness of its use in industry are mentioned and surveyed in [13]. Therefore, in designing a secure embedded system, hardware aspects should also be taken into account. However, few security modeling solutions offer support to cover hardware security requirements in the system. Part of this problem could be due to the need to include concepts for describing hardware in such modeling languages. Specifying hardware units with built-in security support can affect allocation and deployment scenarios and thus is important to be modeled. On the other hand, MARTE already includes basic concepts for modeling hardware platforms and resources in embedded systems as well as allocation and deployment. This again shows potentials of MARTE as a common and unifying framework for adoption and definition of security concepts in embedded systems.

D. Potentials of MARTE

The rich concepts in MARTE for specification of non-functional properties, modeling of time, allocation and platform, and also support for schedulability and performance analysis give MARTE good potentials to answer security issues described above especially regarding the unification problems mentioned in [9] and [10] and offering a single framework. From this point of view, use of MARTE is a promising approach for cross-cutting nature of security aspects and providing a concise model of the system without redundant information modeling. For example, by extending

MARTE with security concepts, one use case could be to define secure ports and connections for components in MARTE component model, and this way highlight secure data flow and encryption requirements in the system and disallow designs that breach security (secure flow of data to an untrusted component). Without using a unified approach, several elements may have to re-appear in different models in order to cover various aspects of the systems; for example one to model security, one to model allocation and so on. Another example in which an extended version of MARTE (with security) proves beneficial is in modeling embedded systems which use hardware that have security support. Also as mentioned above, there are families of boards and microcontrollers which have hardware implementations of cryptographic algorithms. MARTE Hardware Resource Modeling (HRM) profile can be extended to enable modeling of such systems. An interesting work which tries to add security modeling to MARTE is [10]. In this study, necessary concepts for modeling and analysis of resilience (as a security feature) are defined as a profile called Security Analysis and Modeling (SecAM). This suggested profile is based on MARTE profile for Dependability Analysis and Modeling (DAM) [14].

III. Motivation Example

To show the benefits and applicability of an extended version of MARTE (MARTE+Security), we use the automotive example that is mentioned in [15]. This example is a desired use case for electronic payment systems that can be embedded in vehicles. In this case, a vehicle enters a parking garage. The vehicle starts interacting with the garage to receive information on parking cost, notifies the driver about that, and upon exit pays the fee automatically through an authorized third party (e.g. vehicle OEM). These interactions are shown in Figure 1. To design such a system, different aspects such as schedulability, performance, allocation and security should be considered.

To model the security aspects of this system, SecureUML is not enough since it only focuses on access control and lacks modeling support for issues like encryption and decryption mechanisms. Another security profile such as UMLsec [8] can be used which covers a wider variety of security concepts. However, as mentioned before, modeling security aspects in embedded systems separately can be problematic. For example, by introducing such automatic electronic payment system in vehicles, there is now a tighter relation between security and other requirements of the system. One problematic scenario is the relation between schedulability, performance and security. If the security protocols that are used require heavy computations and are not well designed, other tasks in the system may miss their deadlines. It can be from a simple window closing task to automatic braking or central lock system which in the end affect safety requirements of vehicles. An extreme case of this situation could be when the vehicle is busy performing the payment, and the driver needs to close the windows or lock the doors due to a burglary threat inside the garage.

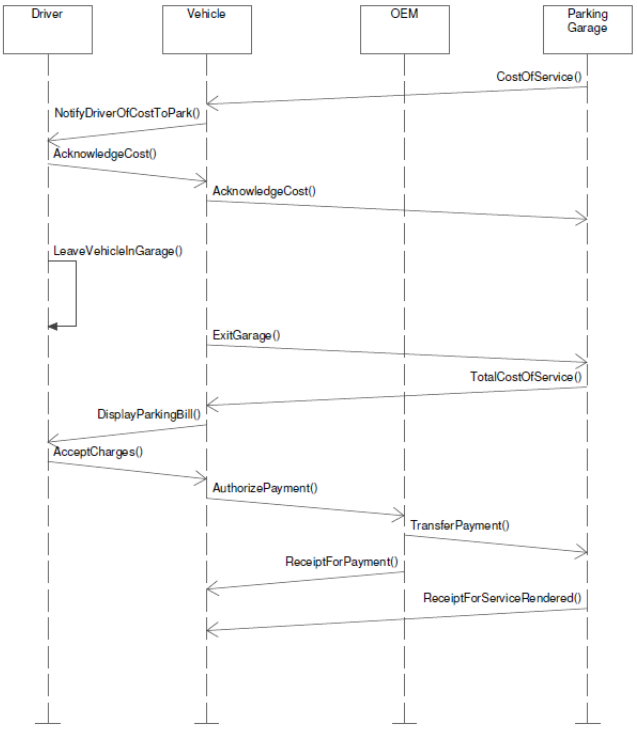

Fig. 1. Electronic payment service for an intelligent parking garage [15]

Also for OEMs to distinguish between vehicles, perform transactions and issue billing information accordingly, a mechanism is required to uniquely identify each vehicle. This mechanism should be immune to impersonation attacks, otherwise malicious parties (e.g. driver of the vehicle) may try to have other people/vehicles' accounts charged.

IV. Modeling Encryption and Enriching Analysis

As discussed, satisfying security requirements comes with a cost in terms of performance, energy and memory consumption and impact on other properties of the system. In order to consider this cost in the design of systems, it is needed to know specifications of applied security mechanisms such as encryption. In this section, we focus on the encryption requirement of *AuthorizePayment()* operation and show how modeling this requirement using MARTE enables performing sensitivity and trade-off analysis.

To model this security requirement we have defined a stereotype called *Encryption* and 'specialized' from it different types of encryption particularly block ciphers that are used in this example. Figure 2 shows definition of this stereotype based on MARTE NFP concepts.

As AuthorizePayment() involves message trasnfer between Vehicle and OEM, it is labeled as CommunicationStep (GaCommStep or SaCommStep depending on the Analysis Context) [4]. First, applying our stereotype, AuthorizePayment() is specified as:

≪BlockCipher≫ AuthorizePayment() {algorithm=AES , blockSize=(128,bit), keySize=(128,bit), rounds=12,

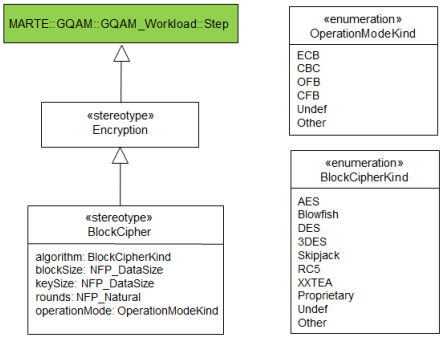

Fig. 2. Definition of *BlockCipher* stereotype

operationMode=ECB}

In a schedulability scenario, *SaCommStep* from Schedulability Analysis Modeling (SAM) profile can then be applied to support schedulability analysis:

≪SaCommStep≫ ≪BlockCipher≫ AuthorizePayment() {algorithm=AES , blockSize=(128,bit), keySize=(128,bit), rounds=12, operationMode=ECB, msgSize=(150,B)}

Having the above details specified about the encryption mechanism in the model enables us now to incorporate the results of studies such as [16] and [17]. In [17], execution times for different encryption algorithms based on input size have been measured. [16] investigates energy consumptions of block cipher encryption algorithms with different settings (e.g. rounds, mode of operation, etc.). It also includes some comparisons of software versus hardware encryption. Figure 3 and 4 show examples of these results.

Input size (bytes)	DES	3DES	AES	BF
20,527	24	72	39	19
36,002	48	123	74	35
45,911	57	158	94	46
59,862	74	202	126	58
69,646	83	243	143	67
137,325	160	461	285	136
158,959	190	543	324	158
166,364	198	569	355	162
191,383	227	655	378	176
232,398	276	799	460	219
Average time	134	383	228	108
Bytes/sec	835	292	491	1,036

Fig. 3. Execution times of block cipher algorithms in ECB mode of operation [17]

These values and derived graphs/tables (for different platforms) can be part of a security-performance analysis tool. Using performance comparisons, results similar to the following example can be achieved:

≪SaCommStep≫ ≪BlockCipher≫ AuthorizePayment() {algorithm=AES , blockSize=(128,bit), keySize=(128,bit), rounds=12, operationMode=ECB, msgSize=(150,B),

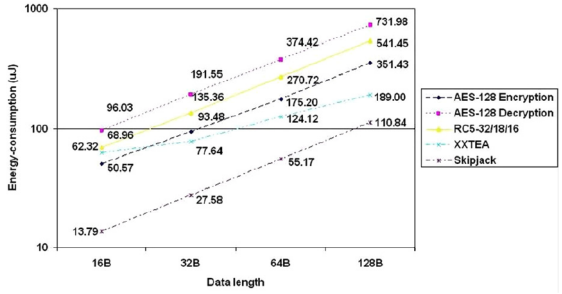

Fig. 4. Energy consumption of block ciphers for different data lengths [16]

execTime=(330,ms,min,calc)}

Since the security-performance analysis only knows about security concepts it only calculates the execution time for the encryption part of AuthorizePayment operation and the total execution time of this operation will be more than this value (hence using 'min' as StatisticalQualifier). In cases where energy consumptions of encryption algorithms are also important such as in Wireless Sensor Networks and in order to perform an energy consumption analysis on the model, energy values can be calculated and included in the model analysis context:

≪GaCommStep≫ ≪BlockCipher≫ AuthorizePayment() {algorithm=AES , blockSize=(128,bit), keySize=(128,bit), rounds=12, operationMode=ECB, msgSize=(150,B), execTime=(330,ms,min,calc), energy(0.23,mj)}

Knowing timing and energy values based on the chosen encryption method and input message size, sensitivity analysis on the model is possible now to determine the best trade-off between security, timing and energy consumption. For example, if it is realized that the execution time (and/or energy consumption) is too much, the message to be encrypted can be reduced in size, parameters of encryption algorithm may be changed or another encryption method can be selected instead. It is important to note that these evaluations are now feasible before implementation and reaching the code level.

V. Vehicle Authentication

In the automatic electronic payment system, in order for the OEM to distinguish between different vehicles, several mechanisms can be used. One way could be to issue a smart card for each owner and have an embedded card reader in the vehicle to read the card information and send it for identification. Another possible solution is to use a unique identifier for each vehicle. For instance, registration plate number of the vehicle can be stored in a memory chip and retrieved and sent over to the OEM to identify the vehicle.

To store unique identification information securely, a secure memory module can be used (refer to [18] for examples of available secure memory modules). To be able to include this scenario in the system, a stereotype for secure memory modules is defined. Building upon hardware resource modeling

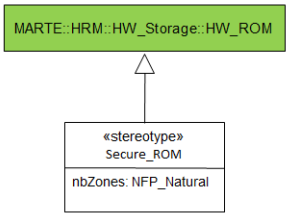

Fig. 5. Hardware Security Modules: Secure ROM

concepts of MARTE, such a memory type can be defined as shown in figure 5. This memory type can be used in allocation and deployment models to emphasize secure storing of information and also enable system designer to apply more secure allocation and deployment scenarios.

VI. Incorporating Extensions in MARTE

One step in adding security to MARTE is to apply an appropriate taxonomy for security concepts as there are different suggested classifications in this area. Definition of security concepts in a way that the model can be used (e.g. through transformation) as input model to security analysis tools is also an important issue that should be taken into account. For example, Encryption concepts we defined in this work can be transformed to Secrecy concept of UMLsec if security aspects of the model are to be analyzed by UMLsec analysis tool.

To offer security concepts as extension of MARTE, we are working on defining an appropriate structure for the profile considering different classification and taxonomies for security that will also be in line with MARTE's way for grouping of concepts into packages and subprofiles. Figure 6 shows the current structure of our suggested profile.

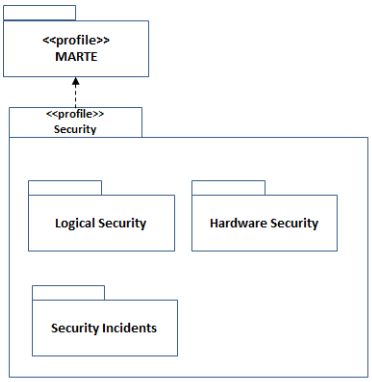

Fig. 6. Suggested structure for Security profile

Logical Security in this structure will consist of the following packages: Secrecy_Confidentiality, Integrity, Authentication, Authorization_AccessControl, Non_Repudiation and DataFreshness. Hardware Security will include concepts for modeling hardware intrinsic security and physical measures to protect and secure embedded systems, e.g. against side channel attacks and making systems tamper-resistant. To group concepts such as vulnerability, threat, attack, attacker and intrusion, Security Incident package is defined which is inspired by studies such as [19].

VII. Conclusion and Future work

There are efforts on defining modeling languages for security, however, when it comes to systems such as in embedded domain where system properties are tightly interconnected, security requirements cannot be considered in isolation from other requirements such as timing and energy consumption. In this paper, we discussed this issue and how MARTE can serve as a common framework to build security concepts upon. Strong features of MARTE toward this goal were highlighted and it was shown how an appropriate specification of security can help to perform sensitivity and trade-off analysis among requirements on the model. The latter is especially important in embedded systems domain where resources are limited.

Including security concepts and dedicating packages and profiles for them in MARTE, can also help with raising the awareness of system designers towards considering and including security decisions in design models. This explicit support for security in the modeling language helps with the problem of considering security as an afterthought and added feature to the system. Especially that you cannot almost build an embedded systems these days with no security.

As a future work it is interesting to try to generate code with security features from MARTE+security models. For example, Encryption and BlockCipher stereotypes and their respective properties can help with determining the right encryption algorithm to implement and generating the code for it. Successful generation of code from these specifications and its generation percentage and coverage is left to be investigated and tested as a continuation of this work. Also using MARTE+security models as inputs to security analysis tools to evaluate security requirements and security level of the system is another future direction of this paper.

References

[1] S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in embedded systems: Design challenges," *ACM Trans. Embed. Comput. Syst.*, vol. 3, pp. 461–491, August 2004. [Online]. Available: http://doi.acm.org/10.1145/1015047.1015049

[2] P. Kocher, R. Lee, G. McGraw, and A. Raghunathan, "Security as a new dimension in embedded system design," in *Proceedings of the 41st annual Design Automation Conference*, ser. DAC '04. New York, NY, USA: ACM, 2004, pp. 753–760, moderator-Ravi, Srivaths. [Online]. Available: http://doi.acm.org/10.1145/996566.996771

[3] S. Gürgens, C. Rudolph, A. Maña, and S. Nadjm-Tehrani, "Security engineering for embedded systems: the secfutur vision," in *Proceedings of the International Workshop on Security and Dependability for Resource Constrained Embedded Systems*, ser. S&D4RCES '10. New York, NY, USA: ACM, 2010, pp. 7:1–7:6. [Online]. Available: http://doi.acm.org/10.1145/1868433.1868443

[4] MARTE specification version 1.0 (formal/2009-11-02), http://www.omgmarte.org.

[5] T. Lodderstedt, D. A. Basin, and J. Doser, "Secureuml: A uml-based modeling language for model-driven security," in *Proceedings of the 5th International Conference on The Unified Modeling Language*, ser. UML '02. London, UK: Springer-Verlag, 2002, pp. 426–441. [Online]. Available: http://portal.acm.org/citation.cfm?id=647246.719477

[6] K. Alghathbar and D. Wijesekera, "authuml: a three-phased framework to analyze access control specifications in use cases," in *FMSE '03: Proceedings of the 2003 ACM workshop on Formal methods in security engineering*. New York, NY, USA: ACM, 2003, pp. 77–86.

[7] K. P. Peralta, A. M. Orozco, A. F. Zorzo, and F. M. Oliveira, "Specifying security aspects in uml models."

[8] J. Jürjens, "Umlsec: Extending uml for secure systems development," in *UML '02: Proceedings of the 5th International Conference on The Unified Modeling Language*. London, UK: Springer-Verlag, 2002, pp. 412–425.

[9] V. Thapa, E. Song, and H. Kim, "An approach to verifying security and timing properties in uml models," in *Engineering of Complex Computer Systems (ICECCS), 2010 15th IEEE International Conference on*, 2010, pp. 193 –202.

[10] R. J. Rodríguez, J. Merseguer, and S. Bernardi, "Modelling and Analysing Resilience as a Security Issue within UML," in *SERENE'10: Proceedings. of the 2nd International Workshop on Software Engineering for Resilient Systems*. ACM, 2010, accepted for publication.

[11] F. Noyrit, S. Gérard, F. Terrier, and B. Selic, "Consistent modeling using multiple uml profiles," in *Model Driven Engineering Languages and Systems*, ser. Lecture Notes in Computer Science, D. Petriu, N. Rouquette, and y. Haugen, Eds. Springer Berlin / Heidelberg, 2010, vol. 6394, pp. 392–406.

[12] P. Pecho, J. Nagy, and P. Hanacek, "Power consumption of hardware cryptography platform for wireless sensor," in *Parallel and Distributed Computing, Applications and Technologies, 2009 International Conference on*, 2009, pp. 318 –323.

[13] Global Semiconductor Industry , "Hardware intrinsic security: Fabless perception & awareness study," http://www.gsaglobal.org/publications/hisreport/docs/HISReport.pdf, October 2010.

[14] S. Bernardi, J. Merseguer, and D. Petriu, "A dependability profile within marte," *Software and Systems Modeling*, pp. 1–24, 2009, 10.1007/s10270-009-0128-1. [Online]. Available: http://dx.doi.org/10.1007/s10270-009-0128-1

[15] K. Prasad, T. Giuli, and D. Watson, "The case for modeling security, privacy, usability and reliability (spur) in automotive software," in *Model-Driven Development of Reliable Automotive Services*, ser. Lecture Notes in Computer Science, M. Broy, I. Krüger, and M. Meisinger, Eds., vol. 4922. Springer Berlin / Heidelberg, 2008, pp. 1–14.

[16] J. Lee, K. Kapitanova, and S. H. Son, "The price of security in wireless sensor networks," *Comput. Netw.*, vol. 54, pp. 2967–2978, December 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2010.05.011

[17] A. Nadeem and M. Javed, "A performance comparison of data encryption algorithms," in *Information and Communication Technologies, 2005. ICICT 2005. First International Conference on*, 2005, pp. 84 – 89.

[18] Atmel CryptoMemory , http://www.atmel.com/products/securemem/, Accessed: February 2011.

[19] J. D. Howard and T. A. Longstaff, "A common language for computer security incidents," CERT: http://www.cert.org/research/taxonomy_988667.pdf, 1998.