Toward Designing an Adaptive Communication Security for the Next-generation Mobile Computing

A. M. Rashwan[1], A-E M. Taha[2], H. S. Hassanein[1] and A. Radwan[3]

[1]Telecommunications Research Lab
School of Computing
Queen's University
Kingston, ON, Canada K7L 3N6
{arashwan, hossam}@cs.queensu.ca

[2]Electrical Engineering Department
Alfaisal University
P.O. Box 5092
Riyadh 11533 KSA
ataha@alfaisal.edu

[3]Instituto de Telecomunicações
Campus Universitário de Santiago
3810-193 Aveiro,
Portugal
aradwan@av.it.pt

***Abstract*— Mobile computing proved to be essential in today's cyber communications. However, entities in mobile computing are known of having limited energy, physical, and logical resources. This imposes various challenges that greatly affect communication quality and performance of those mobile entities, especially when applying computationally-intensive security measures that are essential for protecting the communication sessions. Therefore, it becomes vital to seek suitable security techniques that balance between the communication performance and the resource context of those mobile entities. This paper investigates some possible options toward implementing an adaptive security measures that work with various mobile and next generation Internet entities. The paper basically studies the communication performance of mobile entities when security functions are running on them with and without operating adaptations. While the focus in this paper is about the Message Authentication Code group of security functions, the work can be generalized to include any resource-intensive security measures including both other cryptographic (such as encryption) and non-cryptographic measures (such as challenges).**

Keywords-component; adaptive security; dynamic resource management; message authentication code; message hashing; mobile computing security; next generation Internet security.

I. INTRODUCTION

The success of the Next generation Internet (NGI), including Internet-of-Things (IoT), is based on having entrusted communications between its entities. Therefore, all entities within the NGI must incorporate some sort of security measures that at least ensure the validity and authenticity of the transmitted information. With NGI entities operating on various capabilities and requirements, it is essential to design the prospective security measures to be able to scale and adapt to the communication context and without sacrificing the protection levels they intend to provide.

Achieving a feasible prospective security measure for the NGI means that it should not put a huge burden on the availability of the prospective NGI entities, their hosting entities, and/or the intermediate NGI nodes. Therefore, security measures should only implement the necessary functions to ensure proper communication entrustment (for example, leaving encryption optional for applications). In addition, security measures should utilize mechanisms that ensure lightweight resource demands of the communicating entities. Ensuring lightweight demands will reduce the chances of having resource exhaustion attacks on future entities due to the increased overhead coming from the applied security measure.

While today's communicating entities vary in resource capabilities and requirements, many of the popular security protocol implementations used today are based on non-adaptive functionality that does not consider their context. Examples of such protocols include TLS, IPSec, PGP, and Kerberos, to name a few [1]. With these protocols, there is no mechanism of adapting the data integrity strength or the encryption strength in accordance with the communicating entities context; putting weaker and resource-limited entities into huge burden, and increasing the risk for those entities to go down. In addition, the dynamicity of today's mobile computing environments may cause such security protocols to impact the possibility to achieve acceptable Quality of Experience (QoE) levels due to the inability to adapt the security strength in correlation with the available resources. Therefore, it becomes important to work into designing a security protocol that adapts its security strength based on the context of the communicating entities and within acceptable security sacrifices.

This paper investigates the design requirements toward having an adaptive security measure/protocol that can work with the variety of the NGI entities. Our work focuses on a group of security functions, known as *Message Authentication Code* (MAC), which are used to ensure communication data integrity between entities. In this paper, we study the effect of using MAC functions, with and without adaptations, on the performance of the communication sessions that utilize them. We introduce an adaptive strategy, named *Authentication-Trim*, to adjust the security strength based on processing latency context in reference with lookup tables representing pre-evaluated resource demands. We show the performance of different adaptation schemes and withdraw conclusions of open research challenges and issues.

The remainder of this paper is organized as follows. Section II refers to the background and motivations for investigating possible options toward designing an adaptive security measure. Section III describes the design considerations and recommendations toward having an adaptive security measure. A proposed design for the authentication-trim strategy, with design assumptions and limitations, is illustrated in Section IV. Section V presents the performance comparison when running the proposed adaptive strategy verses non-adaptive and randomly adaptive ones. Open issues and concepts are discussed in Section VI. Finally, conclusion and future directions are mentioned in Section VII.

978-1-4799-6664-6/16/$31.00 ©2016 IEEE

II. Background And Motivation

Designing an adaptive security protocol to address resource dynamic context is not new in the literature. Examples of earlier software-based efforts include Adaptive SSL [2], ACSA [3] and RSSA [4]. Adaptive cryptographic accelerator [5] is an example of hardware-based efforts. Another example of an adaptation effort is the usage of flexible policies enforcement with an adaptive risk/challenge management engine [6]. We can categorize the existing security adaptation efforts based on their key goals as follows:

- Optimizing the process by selecting security functions based on the available resources and the requirements.
- Increasing the resistance against attacks through enforcing adaptive policies, challenges, and security steps based on the communication context and requirements.

The key issue for most existing security adaptation efforts is the consideration of stand-alone physical entities; leading to the reliance on evaluating absolute resource demands for adaptation (such in ACSA or RSSA). Unlike existing communication entities, the NGI entities will have different characteristics as shown in the following:

- **Entity's Nature.** The current computing trend of the Internet is going mobile and cloud, and so NGI entities can be any object that is able to communicate: (e.g. an application, a sensor element (as in IoT), a self-publishing file, a virtual computer, a smart phone,…etc). Entities in this case may not have physical resources of their own but shared with other entities within NGI. Moreover, the same entities, due to mobility, may have different physical capabilities during their communication sessions. Therefore, designing an adaptive security measure based solely on the determination of physical resources demands will be challenging and difficult to standardize.
- **Entity's Availability.** Since NGI entities may no longer have dedicated resources, they are more vulnerable to the resource exhausting attacks; increasing their availability risk. In addition, sharing physical resources between entities impose additional challenges in isolating security attacks; affecting availability for entities even further. For a prospective adaptive security measure to operate effectively in NGI, it should be designed not to cause significant overhead that worsen the availability, yet operate in a sufficiently secure manner. This may include the utilization of easy-to-compute yet context-reflecting decision factors in the adaptation process. In addition, we believe that a strong security measure is not about how much time it takes to break, but it is about how much effort it will take from the attackers to break. For example, a weak and fast cryptographic function, with short key refreshment intervals can be as secure as using a very strong and slow cryptographic function, and without having excessive load on the involved entities.

It is clear that any prospective adaptive security measure must be wisely designed while considering the new NGI entity concept in quest of providing good security with least possible overhead. In the following sections, we go through the details toward designing an adaptive security measure for the NGI.

III. Designing an Adaptive Security Measure for NGI

In the following subsections, we discuss the design criteria and the recommended components for having an adaptive security measure for the NGI.

A. Design Criteria

- **The Available Security Functions at Entities.** Not all entities will have access to the same set of security functions. For example, low-powered entities, such as sensors, will mostly utilize a single security function, losing the ability to adapt the operation from a set of functions and so the ability to communicate securely with some other entities. Designing an adaptive measure must take into consideration the variance of the available security functions. Limited-power entities may utilize a limited adaptive measure with the ability to communicate through some sort of security proxy with other entities that do not share the same security function availability.
- **The Security Functions' Resource Demands.** We note that different security functions require different resource demands from the same entity, and that the same security function requires different resource demands from different entities [7]. Pre-determining the computational trends for those functions can be essential for achieving a low overhead decision making process and therefore a low complexity adaptive security measure.
- **The Security Function's Implementation.** Hardware-based security functions do not rely on the entity's main processing capability, but they are usually designed to process a single request at a time. Software-based security functions, on the other hand, can process simultaneous requests but rely on the entity's main processing capability. The design of an adaptive security measure must consider not only resource demands, but also other processing limitations associated with various implementations, such as the additional I/O latency accessing the external security processor, and the number of requests that can be handled concurrently.
- **The Resource-Context of Entities.** Several mobile computing systems are equipped with hardware-based or software-based energy conservation approaches, intended to provide a balance between the power consumption and the computational performance. In situations such as running high on temperature or low on available battery power, entities may reduce their computational ability by turning down the clock of their processors, or by shutting down some of their processing cores [8]. This in turn will increase the latency of executing tasks (including security ones). An adaptive security measure that aims to meet certain performance guaranties should be able to keep up with the dynamicity of the communicating entities.
- **Adaptation Strategy Context.** An adaptive measure that uses network-based strategies, such as using the network utilization, security policies, or risk indicators in the adaptation decision, may offer better overall security. However using network-based strategies may also impose high complexity on the communicating entities. Entity-based strategies, on the other hand, have high chance of

TABLE I
AUTHENTICATION-TRIM CALIBRATION LOOKUP TABLES

Functions	Message Size (Bytes)
64-bit MACs 128-bit MACs 256-bit MACs 512-bit MACs	Apparent Processing Time (ordered low to high)

forgery and break attacks due to the absence of the network's oversight. Prospective adaptive security measures should be designed with a strategy that balances between the entity and the network. We believe that a good design should be entity-based and network-assisted. Networks in such case can assist the adaptive operation through monitoring and informing related entities with anomalies and security breaches.

B. Recommended Components for a General-Purpose Adaptive Security Measure

Based on the aforementioned design criteria, we propose the following two components to facilitate implementing a general-purpose adaptive security measure.

- **Evaluation Module (Calibrator).** The aim of this module is to take care of the variance of determining the computational trend for a same security function under different entities; simplifying the adaptation process. Therefore, it is responsible to generate offline lookup tables with metrics that represent the computational and latency trend demands of the incorporated security functions under different pre-determined context scenarios. The lookup tables are then used to help in selecting the suitable security function during the adaptation process. Delay-based metrics can be considered good representations to use as they do not usually require the knowledge of physical resources; making them suitable for roaming entities that do not have resources of their own.
- **Adaptation Module (Selector).** The aim of this module is to take care of the runtime variance due to the non-deterministic environmental context that affects the available computational resources to the communication sessions. Therefore, it adjusts the selection of the suitable security function at runtime based on the entity's context and with reference to the trend lookup tables generated by the calibrator. Since the selector runs along the communication session, its decision maker should rely on easy-to-compute and low-overhead fitness values that can be referenced to the aforementioned lookup tables. Such fitness values will be computed primarily based on the metric used in generating the trend lookup tables, and may be subject to constraints and requirements imposed by the communicating entities and the network.

IV. AUTHENTICATION-TRIM ADAPTATION STRATGY

In this section, we propose a delay-based adaptation strategy, named *Authentication-Trim*, that utilizes the aforementioned design considerations to achieve a general-purpose adaptive security measure to use in NGI. A delay-based metric, representing the processing time, is used in generating the calibrator's trend lookup table. The selector module utilizes a fitness value that represents the computational power trim behavior, where the selector tends to use less demanding functions when the fitness value indicates high resource demands and vise versa. The following subsections illustrate the implementation details of the Authentication-Trim strategy for the use with MAC functions.

A. Evalaution Metrics

In this paper, we only consider the MAC tagging operation. Therefore, the metrics involved are:

- **Message Size.** For the message to be tagged (in bytes).
- **MAC Size.** The size of the verification tag generated for a given message (in bits).
- **Resource Demand**. The resource needed for tagging a given message.

In this paper, we consider using the "*apparent processing time*" as the resource demand metric. The metric, which is variation from the one in [7], represents the time required (in seconds) to tag one message of size *n*. This delay-based and easy to compute will be suitable to evaluate the computational trends of MAC functions for various entities without the need for the complex interactions to probe physical resources.

B. The Calibration Proccess

The Authentication-Trim strategy requires the invocation of the calibration module in cases where there is a change on computational trends of the security functions on a given entity such as the changes on the entity's hardware or firmware. In case of roaming logical entities, the calibration module of those entities will just inherit the lookup tables (adjusted with the interfacing latency) from their hosting entities.

On physical entities, the calibrator generates the trend lookup table using a single execution core running on its middle clock frequency. This is due to the fact that modern computing systems do not operate on the highest frequency nor use all the execution cores all the time in order to conserve energy [8]. Anyway, finding the best resource setup for the calibration process is not the aim of this study.

Table I represents the general format of a lookup table generated by the calibrator in this study. The MAC functions are sorted into groups; ordered by their MAC size that we assume it roughly represents the MAC function strength. For each group, the functions of the same MAC size are sorted according to their apparent processing time (from low to high) for the given message size. It is clear that in this lookup table, it is not necessarily that the MAC function with the weakest strength will have overall lowest apparent processing time and vise versa. However, using weaker MAC functions means less message size overhead, thus the sorting based on tag size. As with the calibration resource setup, finding the best lookup table organization is not the aim of this study.

The number of lookup tables and the selection of message sizes are greatly dependent on how well the tables should represent the computational trends. Very few tables may lead to an inefficient adaptation, while having large number of tables requires extensive calibration time and storage capacity. In this study we based the selection of message sizes and the number of tables based on the trending charts in [7], generating 6

TABLE II
THE AUTHENTICATION-TRIM ADAPTATION ALGORITHM

```
Sample size/time consumed for n messages
If n samples collected or time elapsed then
    If fitness_average_time < 0.67 trend_time then
        SAT++
    Else If fitness_average_time > 1.33 trend_time then
        SAT--
    Else
        If SAT > 0 then
            SAT--
        End If
        If SAT < 0 then
            SAT++
        End If
    End If
    If SAT ≥ 30 then
        Switch to a stronger available engine (Px = Py) and adjust
        LAT, provided that:
        1) there is a stronger engine (Px) to select,
        2) Py supported by the receiving entity,
        3) and no constraints limiting the power up
    End If
    If SAT ≤ -30 then
        Switch to a weaker available engine (Px = Py) and adjust
        LAT, provided that:
        1) there is a weaker engine (Px) to select,
        2) Py supported by the receiving entity,
        3) and no constraints limiting the power down
    End If
End If
```

lookup tables representing message sizes between 68 bytes and 65535 bytes (IPv4 Path MTU limits) [1].

C. The Adaptation Proccess

The adaptation process of the MAC security strength is analogous to the automotives' On-Board Diagnostics (OBD) fuel management system [9]. The strength and resource demands of a MAC function resemble the amount of fuel injected into the engine per time unit. The OBD fuel management system maintains two fuel trim meters, which are used to control the amount of fuel injected in each cycle to maintain engine's efficiency. Similarly, the proposed adaptation strategy uses two indicators to determine the resource demand of the currently selected function and the demanded adaptation direction. The first meter is named Short-term Authentication-Trim (SAT), and represents how far the active MAC function's resource demands from the trending obtained from the calibration lookup tables. The second meter is named Long-term Authentication-Trim (LAT), and represents the strength of the selected MAC function.

Table II illustrates the selection process. First, the process starts along with an associated communication session with a selection of pre-determined MAC function (In this study, the pre-determined MAC function is the one used by TLS, known as HMAC-SHA1 [10]). The fitness value represents the average tagging time required per message for *n* samples. The value of SAT is initialized to 0, while the value for the LAT is initialized to the position of the selection MAC function at the startup. If the fitness value goes below the 0.67 of the time observed from the lookup tables, the SAT value is incremented by 1. If the fitness value goes above the 1.33 of the time observed from the lookup tables, the SAT value is decremented by 1. Once the SAT hits -30 or 30, the selection process will "power down" or "power up" accordingly; selecting a weaker or a stronger MAC function that leads the SAT to return to 0. If there are constraints or requirements, such as minimum throughput and minimum security strength, that prevent powering up or down, then the adaptation process will do nothing. The SAT in this case will keep in the negative "rich" range indicating high resource demand, or in the positive "lean" range indicating low resource demands.

D. Assumptions

- To simplify the analysis we do not currently consider involving keying demands in the adaptation strategy.
- For the message sizes that do not have lookup tables, the apparent processing time is estimated using a linear interpolation/extrapolation from the existing lookup tables.
- The adaptation is a simple send-only strategy. The receiving entity will inform the sending entity if the selected function is supported or not.
- The 67% and 133% fitness thresholds are based on the performance gain observed in [7] with the Simultaneous hardware Multi-Threading (SMT) (33%). Researching the best threshold is out of the scope of the paper.
- The interval for the key refreshment of each MAC function depends on its resistance to key breakage. Determining such interval is not in the scope of the study and left for future investigation. For the simplicity of the analysis, we currently consider a fixed interval that is correlated to the MAC size.

V. PERFORMANCE ANALYSIS

To study the performance of the proposed Authentication-Trim strategy, we conducted several experiments comparing it with a non-adaptive strategy and with a randomly adaptive strategy. The following subsections present the analysis of the results obtained from conducting the experiments.

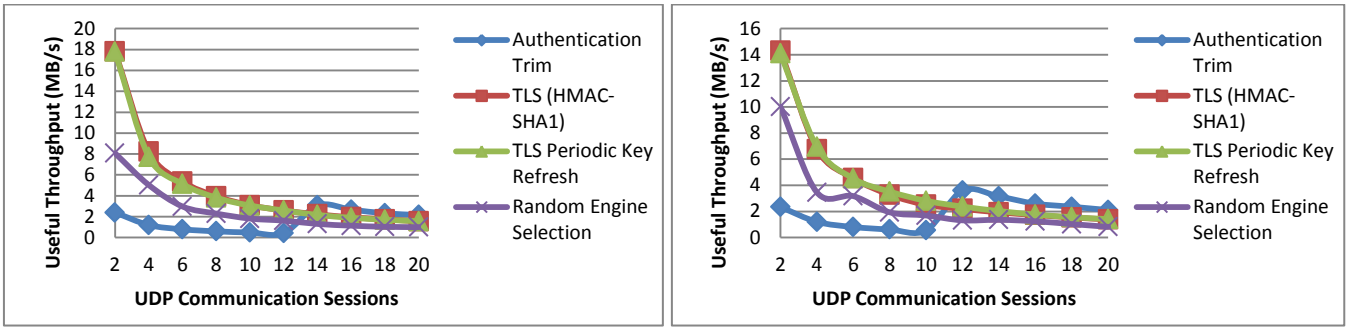
A. Performance of Authentication-Trim Strategy

Fig. 1 shows the useful throughput of running streaming sessions using UDP with message size of 1460 bytes. It is noted that the authentication-trim strategy performs well only with high number of sessions. There were no constraints to the power up adaptation, and so the authentication-trim strategy kept powering up the MAC strength to the maximum; affecting the available resources for data transmission.

It is also observed from the processing breakdown of the communication operations (refer to Table III) that the authentication-trim strategy does not add substantial overhead (represented by the function selection time + key setup time) when compared to the time required for tagging. This is expected as using lookup tables and simple metrics should not demand intensive computation demands.

B. Using Periodic Key Changes

In Fig. 1, both the non-adaptive HMAC-SHA1 strategy and the HMAC-SHA1 with periodic key refreshing strategy performed nearly the same. This is a strong indication to the fact that that key refreshment time (at the entity) requires a very negligible time compared to the time spent in tagging



Texas Instruments' OMAP 4460 ARM Dual-core Cortex™ A9 (0.7 Ghz) Intel Atom N270 (1.33Ghz)

Fig. 1. Useful Throughput using different adaptation approaches to apply MAC security measures

TABLE III
PROCESSING PERCENTAGE BREAKUP FOR DIFFERENT STRATEGIES
TEXAS INSTRUMENTS' AM3358X ARM CORTEX™ A8 – 1GHZ – 20 SESSIONS

Strategy	Message Formatting	Function Selection	Key Setup	Tagging	Transmission
Authentication-Trim	3.19349%	0.01265%	0.04603%	94.34950%	2.39828%
TLS (HMAC-SHA1)	2.46583%	0.00000%	0.00001%	95.58910%	1.94504%
TLS with Periodic Key Refresh	2.82068%	0.00000%	0.00184%	95.06540%	2.11209%
Random Selection	1.86120%	0.00282%	0.00021%	96.70500%	1.43078%

messages (refer to Table III). Therefore, key refreshment can be used to enhance security via making it more resistant to brute force attacks and without significantly increasing the processing overhead.

C. *The Use of Random Strategy*

It is observed that the random strategy does not add computational benefit as opposed to the use of a deterministic strategy. In fact, the randomness nature has led to having worse unfairness levels between communication sessions with Jain's Fairness Index down to 0.87 compared to 0.98 with other strategies (1.0 represents perfect fairness).

D. *The Impact of Power Management on The Authentication-Trim Strategy*

Modern mobile systems utilize different power management approaches to conserve energy and reduce operating temperature, with most of these techniques leading to either reducing the processor clock speed/voltage or shutting down some of the processing cores [8]. Based on that assumption, we conducted controlled evaluation with different frequency and available core settings.

Fig. 2 shows an example of how the power management can affect the strategy performance. As was expected from a delay-based selection strategy, the tending to power down becomes stronger when either the processing frequency or the number of active processing cores drops. Therefore, good power management techniques can supplement the delay-based authentication-trim strategy in reducing the overhead and improve the communication performance while maintaining security in cases, where temperature and battery capacity are in critical conditions.

VI. OPEN CONCEPTS/ISSUES

Regardless of the adaptation strategy used, there are several open issues that must be considered when operating any of the adaptation strategy with communication sessions.

A. *Overpowering a Communication Session*

It is noted from the analysis that the Authentication-Trim strategy performs poorly with low number of sessions due to its tendency to power up the strength to take advantage of the available resources (based on the delay-based fitness value). This makes the strategy not acceptable in scenarios where high communication performance is desired (such as in audio/video streaming). This does not conclude that the proposed delay-based metric is not a good choice, since the power up issue can be resolved by applying some constraints that exclude the non-complying security functions from the selection process.

B. *The Fitness Value Sampling*

In our evaluation, we assume fixed-number of samples and use their average to compute SAT. However, determining a proper and lightweight sampling process requires considering several factors including the communication link utilization, the crypto-analysis of the involved security functions, and the amount of processing abnormalities that can affect the sampling process.

C. *Transmission Overhead*

Adding tags to message increases the amount of data to transmit and so reduces the available bandwidth for the useful data, especially in power ups. If a communication session utilizes small-sized messages, such overhead is considered problematic, especially if the default nature of the communications within the involved network utilizes small-sized messages. Reduction options to investigate include the selection of security functions suitable for small messages, and combining small messages into bigger messages before using a security function on them. Another option, in case of MAC functions, is to use a truncated MAC with the message, which will increase the MAC collisions and breakages probabilities.

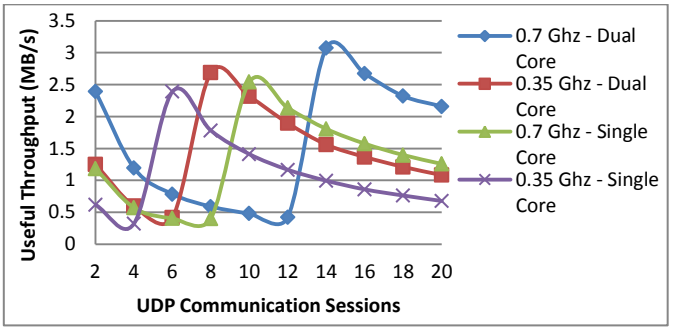


Fig. 2. Useful Throughput under Different Power Conservation Modes - Texas Instruments' OMAP 4460 ARM Dual-core Cortex™ A9

D. *Entities with Limited Capability*

It is noted earlier that some entities, including those used in sensory and embedded networks, have limited ability to use security functions and may not require the implementation of an adaptive security measure. However, those entities may implement the adaptation process if the communication requirements enforce certain security levels. The entities, in such a case, will borrow both the calibration lookup tables and the associated resources from nearby capable aiding entities. This also can be considered an open research issue as part of designing a general-purpose security framework.

E. *The Control Exchance Channel*

In this study, we do not consider how the security keys are negotiated between communicating entities. However we note that in most of the existing security protocols, the negotiation of security keys and other parameters are done using the same communication channel as with the data. Even with an adaptive measure improving security via frequent change of functions and keys, exchanging control information along with the data increases the ability for the attacker to break security measures as opposed to using separate data/control channels. Anyway, designing an adaptive security measure must consider various factors when adopting the separation of control and data channels. Such factors include how the communicating sessions will be managed with the adaptive security measure in place at the entities, and how secure low-overhead control channels can be established and maintained between those entities.

VII. CONCLUSION

In this paper, we investigate different options toward implementing an adaptive security measure that fits various mobile and next generation Internet entities. We observe that a secure adaptive measure can be achieved through a combination of two main techniques. The first technique is to utilize a simple and informative adaptation decision factor in the adaptation process for secure yet lightweight security measure regardless of the entity's nature; hence the adaptation process. The second technique is incorporating the use of weak security functions with short key refreshment as a measure of achieving high communication performance yet having some good resistance against breakage attacks as a result of frequent key changes.

We also found that designing a lightweight adaptive security measure that fits various entities comes with many open challenges, such as overpowering and having high transmission overhead. However, resolving issues such as overpowering can be simply achieved via simple optimization techniques such as incorporating constraints and secondary factors in the adaptation decision making.

Finally, we conclude that it is not feasible to incorporate a unified general-purpose adaptive security measure; especially that many of the communicating entities do not have the ability to operate different security measures. In such scenarios, those entities may benefit from using aiding entities, and security proxies to be able to communicate with other entities securely.

ACKNOWLEDGMENT

This research is funded by a grant from Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] *Internet Official Protocol Standards. RFC 5000, IETF,* May 2008.

[2] C. J. Lamprecht and A. P. A. v. Moorsel., "Runtime Security Adaptation Using Adaptive SSL," in *14th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC '08)*, 2008.

[3] D. Balenson, D. Carman, M. Heyman and A. Sherman, "Adaptive Cryptographically Synchronized Authentication (ACSA): Model and analysis," Glenwood, MD 21738, 1998.

[4] A. V. Taddeo and A. Ferrante, "Run-time Selection of Security Algorithms For Networked Devices," in *ACM Q2SWinet '09*, Tenerife, Canary Islands, Spain, 2009.

[5] Y. Hasegawa, "An adaptive cryptographic accelerator for IPsec on dynamically reconfigurable processor," in *Proceedings 2005 IEEE International Conference on Field-Programmable Technology*, Singapore, 2005.

[6] "RSA Adaptive Authentication Datasheet," January 2013. [Online]. Available: http://www.rsa.com/products/consumer/datasheets/6559_AA_DS_0511.pdf.

[7] A. M. Rashwan, A.-E. M. Taha and H. S. Hassanein, "Characterizing the Performance of Security Functions in Mobile Computing Systems," *IEEE Internet of Things Journal,* vol. 1, no. 5, pp. 339-413, October 2014.

[8] S. Mittal, "A Survey of Techniques For Improving Energy Efficiency in Embedded Computing Systems," *International Journal of Computer Aided Engineering and Technology,* 2014.

[9] D. Cope Enterprises, "OBD II and LDV I/M Programs," April 2004.

[10] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2,* 2008.