Trading off Latency Against Security in Open Energy Metering Infrastructures

Gundars Kalns
Linköping University
Department of Computer and Information Science
SE-581 83 Linköping, Sweden
gunka293@student.liu.se

Simin Nadjm-Tehrani
Linköping University
Department of Computer and Information Science
SE-581 83 Linköping, Sweden
simin.nadjm-tehrani@liu.se

Maria Vasilevskaya
Linköping University
Department of Computer and Information Science
SE-581 83 Linköping, Sweden
maria.vasilevskaya@liu.se

Embedded devices are expected to transform the landscape of networked services in many domains, among them smart homes and smart grid systems. The reliable and optimised operation of smart grids is dependent on reliable data provided by end nodes (e.g. smart meters), and assurance of secure communication across networks. Understanding whether advanced security building blocks have a role to play in forthcoming infrastructures needs a basic understanding of each potential building block with respect to resource usage and impact on timing. In this paper we study the performance penalty of asymmetric cryptography techniques used for protection of wirelessly transmitted data in a prototype smart metering system. The prototype system is built using hardware and software components from "Open Energy Monitor" project using a wireless data link between the metering device and the data collector device. We investigate the use of the Elliptic Curve Integrated Encryption Scheme (ECIES) in two versions — with standard building blocks and with added Elliptic Curve Digital Signature Algorithm (ECDSA) support. The use of the ECDSA allows the system to achieve the non-repudiation property. We compare those cryptographic techniques with the Advanced Encryption Standard in Galois Counter Mode (AES-GCM) technique in two versions — with 128 bit and 256 bit keys. Performance is compared in terms of execution time of (1) preparing data, (2) unpacking it, and (3) roundtrip time. We then discuss the implications of the measurements, where the roundtrip time of sending one measurement ranges from 378 ms in case of AES128-GCM to 16.3 sec using ECIES with ECDSA.

Smart meter infrastructure security, Elliptic Curve Cryptography, Performance and latency trade-off

1. INTRODUCTION

Embedded devices are expected to transform the landscape of networked services in many domains, among them smart homes and smart grid systems. The reliable and optimised operation of smart grids is dependent on reliable data provided by end nodes (e.g. smart meters), and assurance of secure communication across networks. Understanding whether advanced security building blocks have a role to play in forthcoming infrastructures needs a basic understanding of each potential building block with respect to resource usage and impact on timing. Luan et al. (2015) state that "The data [from end nodes] offer utilities many opportunities to apply data analytics to potentially enhance their operational efficiency. For instance, smart meter data can be used for enhancing and estimating voltage and Volt-Ampere Reactive (VAR) optimization benefits, evaluating distribution line losses, identifying and quantifying energy thefts, and enabling improved load forecast, outage management, and distribution system analysis". These applications justify the need for integrity of received data and reliable communication.

Deployment of smart metering systems is associated with huge costs since there are many components that need to be modernised. Smart meters and their installation constitute the biggest part of costs (ICCS-NTUA,ÅF Mercados EMI 2015). A possibility that may be explored to reduce costs is using meters built on low cost and low performance hardware, open source software components, and using standard wireless communication between

© *The Authors. Published by BISL.*

meters and data collectors to gather data. The wireless capturing of the sensor data may then rely on existing communication infrastructures to pass the data further from collectors to utilities. Using open source components also makes the systems subject to large scale verification and quality assurance. However, long term investments in developing such infrastructures needs to rely on the ability of these end nodes to keep up with the emerging security, privacy, and performance requirements.

Digital communication channels are inherently insecure. Security features, however, can have high computational and memory footprint especially on low performance hardware. It is therefore interesting to (1) evaluate the feasibility of integration of open source standard building blocks for security and (2) evaluate their impact on the performance on a baseline (insecure) infrastructure, e.g. in terms of introduced latencies.

In this paper we study the timing overhead of asymmetric cryptography components based on Elliptic Curve Cryptography (ECC) in the context of an open platform for smart metering devices. In particular, we investigate Elliptic Curve Integrated Encryption Scheme (ECIES) with standard building blocks, protecting data confidentiality and integrity, and alternative ones, achieving also non-repudiation property. We perform a comparative performance evaluation contrasting those methods with Advanced Encryption Standard in Galois Counter Mode (AES-GCM) authenticated encryption.

We show that it is feasible to have a smart metering system built on a low cost and low performance hardware platform with integrated ECIES encryption scheme. Our results of using such a platform and having strong cryptography components can be useful in the context of future deployments.

We begin by presenting related works in Section 2 and go on to describe the necessary background in Section 3. Section 4 is devoted to a description of our created software architecture. Performance study and outcomes are presented in Section 5. The paper is concluded in Section 6 with some directions for future work.

2. RELATED WORK

Cleveland (2008) gives an overview of security requirements of the smart metering infrastructure and security threats that could compromise its security. Kaplantzis and Sekercioglu (2012) list security requirements and threats in smart metering systems and highlight some of the effects of malicious hacking activities within the communication network. Energy theft possibilities are studied by McLaughlin et al. (2010). They conclude that new attack vectors appear with modernisation of the grid. Hence, metering devices must be well protected before massive deployment takes place.

One of the standard security measures for devices operating in a digital communication environment is cryptography. Properties of asymmetric cryptography makes it desirable for use in smart metering infrastructure due to the ability of each party to have their own private key. On the other hand, it has higher computational requirements which can make noticeable performance loss in computationally constrained devices. Gura et al. (2004) compare the performance of two asymmetric cryptography algorithms, namely Rivest-Shamir-Adleman (RSA) and ECC on 8 bit CPUs, showing that ECC is better suited for devices with restricted computational resources. Gupta et al. (2005) present results of implementing a web server with Secure Socket Layer (SSL) on constrained embedded devices. They use ECC based cryptography and the device having 8-bit CPU and 4 kB of RAM can complete a full SSL handshake in less than 4 seconds.

Smart meter authentication including the use of asymmetric cryptography is studied by Foudah et al. (2011). They propose a mechanism based on Diffie-Hellman key agreement and hash based Message Authentication Codes (MAC) which they compare with the ECC based mechanism. Compared to our work here, their focus is on the network gateway ability to process messages coming from smart meters.

Molina-Markham et al. (2012) address a similar question to that posed in this paper - i.e. how big is the impact of cryptographic security measures on the performance of smart metering devices. They perform two experiments. One of them is comparison of execution times of commitment and digital signature schemes (not based on ECC) on different architectures. The second experiment is used to compare those schemes with the ones based on ECC on the MSP430 Microcontroller Unit (MCU) platform. This implemented prototype using ECC is capable of producing readings every 10 seconds. In our work, however, we use ECIES encryption scheme and our platform is more constrained than the one they use.

Other approaches of securing smart metering infrastructure include Intrusion Detection Systems (IDS). Raciti and Nadjm-Tehrani (2013) propose an architecture for embedded anomaly detection in smart meters and create an instance of a clustering-based anomaly detection algorithm in a prototype

meter. Tudor et al. (2015) investigate the need to monitor encrypted traffic in a smart metering infrastructure. They propose a methodology for an encrypted command recognition component, as part of an IDS for the metering infrastructure. A multi-agent based simulation of smart meter networks is studied by Lang-Muhr et al. (2015). The simulations of systems in regular operation can be used to deduce whitelist descriptions of network behaviour. An active area of research is the topic of key management in large scale deployments like a metering infrastructure, e.g. Das et al. (2012), and needs a special attention, though outside the scope of this paper.

Although studying the cost of hardware designs for security is subject to meticulous studies (see e.g. Good and Benaissa (2008)), the resource costs of adding security implemented in software is much less studied. A relatively early study of the resource costs was in the context of adding security mechanisms to tactical networks (Matt 2005). Lake et al. (2014) state that "the biggest challenge from the device side is that a lot of M2M/IoT devices do not have enough capability to do the encryption on the device."

In our work we use the open smart meter platform which is a more constrained hardware than platforms in earlier works. To the best of our knowledge there haven't been any studies of the performance and security trade-off particularly in the open smart metering platforms. In addition, there is a lack of studies where ECIES encryption scheme is used in the context of smart metering systems.

3. BACKGROUND

In this Section we provide an overview of the two main ingredients of the experimental platform that we have built: The technology base for the metering infrastructure, and basic cryptographic building blocks.

3.1. "Open Energy Monitor" platform

"Open Energy Monitor" (Open 2016) is an open source project with a goal to develop home energy monitoring systems. Those systems can be used to monitor real-time energy consumption as well as to see detailed history. The hardware developed in "Open Energy Monitor" project consists of two devices: the meter and the data collector.

The meter is a device based on ATmega328 MCU with a RFM12B wireless radio module capable of measuring current and voltage, calculate power, and transmit this data over the wireless channel. The main limiting factors of the MCU performance include its 8bit architecture, 16 MHz operating frequency, 2 kB of RAM, and 1 kB of non-volatile EEPROM memory. Additionally, the limiting factor of RFM12B wireless radio module is a maximum packet size of 66 Bytes. The metering device with four non-invasive clip-on current sensors and an AC adapter with integrated voltage sensor is depicted in Figure 1. In this work, however, we use one current sensor since it is enough for emulating the smart metering application. The firmware of the metering device runs baseline code written in the C language. Its functionality is to compute energy consumption by numerically integrating calculated power values through interrupt-based routines.

Figure 1: *"Open Energy Monitor" metering device (Open 2016).*

The data collector device, which receives measurements from the meters through wireless channel, is based on Raspberry PI minicomputer with a RFM12B radio module. It runs the Raspbian operating system with web server software executing "Open Energy Monitor" code, written in the PHP language. When the user connects to data collector device using an Ethernet port, current and past energy consumption data can be viewed through a specially developed web-application.

3.2. Elliptic curve cryptography and ECIES

Elliptic Curve Cryptography (ECC) is an asymmetric cryptography technique, known to be more lightweight than other asymmetric cryptosystems, such as RSA cryptosystem in terms of computational and memory needs. These characteristics of ECC make it a suitable choice for the resource constrained architecture of our energy metering device.

We explored how to protect the confidentiality and integrity of every measurement and command with asymmetric encryption, and since ECC cannot be directly used to encrypt and decrypt data, we chose Elliptic Curve Integrated Encryption Scheme (ECIES). Integrated encryption schemes consist of building blocks such as symmetric encryption, hash function, Message Authentication Code (MAC)

scheme, key derivation function, and key agreement function. The key agreement function of ECIES is based on one-pass Elliptic Curve Diffie-Hellman (ECDH) algorithm which uses one static and one ephemeral key pair. For all ECIES building blocks a choice can be made between different algorithms. There exist at least four different standards, each defining a different set of supported algorithms (Martinez et al. 2010). One of those standards is SEC 1 (Certicom Research 2009), and since it was the only one available to us, our choices were based on this standard. ECC requires an agreement on a number of domain parameters such as finite field, curve parameters, cofactor value, base point, and its order. To foster interoperability SEC 1 recommends to use standardized sets of domain parameters specified in the SEC 2 (Certicom Research 2010) standard.

Asymmetric cryptography has properties that can be used to achieve non-repudiation, which is considered to be an important property for smart metering systems (Kaplantzis and Sekercioglu 2012). The ECIES scheme, however, does not support it. The ECC-based algorithm that can be used to achieve this property is the Elliptic Curve Digital Signature Algorithm (ECDSA).

4. EXPERIMENTAL SECURE METERING PROTOTYPE

Since the functionality of the "Open Energy Monitor" system is too basic, we began to redesign the software in order to bring it closer to real life smart metering applications. This implied making it more feature-rich, but also more reliable and secure. The process included requirement specification, design of software architecture according to those requirements, and integration of our chosen cryptography methods so that data transfer between meter and data collector can be protected.

4.1. Functional and non-functional requirements

By reviewing information in Open meter standard (Open Meter 2009), supported by several industrial stakeholders, and the European SecFutur research project (The SecFutur project 2010) which aimed to support the development of dependable and secure embedded systems, we chose the following requirements for our system. The system should support:

1. Configuration, calibration, registration and time synchronization capabilities through administrator commands. Since we don't use wired communication, all data, including commands, are delivered over the wireless channel.
2. Timestamp for every measurement with a granularity of a second. Time of the meter is synchronised with the data collector.
3. Acknowledgements for messages received for both the meter and the data collector. Also resending messages in case an acknowledgement is not received.
4. Possibility for the meter to store multiple measurements in memory in case of not being able to successfully forward the data to the collector.
5. Usage of non-volatile memory for storing measurements in the meter so that in case of a restart all unsent data are not lost.
6. Ability to return to normal execution in case of unexpected restart. In such a case an alert message is sent to the collector.
7. Protection of confidentiality and integrity of data in memory and during transfer.

These requirements seem to be simplistic and would appear to be a minimum baseline. Nevertheless we consider them as adequate for operating the metering system while performing the experimental studies on latency and security trade-offs.

4.2. Overview of metering functionality

An overview of the basic functionality of the metering device with addition of encryption/decryption elements is shown in Figure 2. We use EEPROM memory as a non-volatile storage for measurements and based on the requirement to protect data both in memory and transmission, we encrypt data before writing to it. Energy consumption measurements are based on software interrupts because of the necessity to integrate power measurements to get the total consumption. The update of this value therefore takes place regularly, and during measurement collection state it is just fetched from memory.

In a similar fashion we designed the software for the data collector device with features for sending commands, acknowledgements, and data encryption and decryption capabilities.

4.3. Integration of security building blocks

In this work we investigated the ECIES scheme with standard building blocks as well as the modified version with ECDSA support (noted here as ECIES_ECDSA). As for the security strength of algorithms used in ECIES we chose 128 bits, following NIST recommendation SP 800-57 (National Institute of Standards and Technology 2016), which states that 128 bits is the minimum

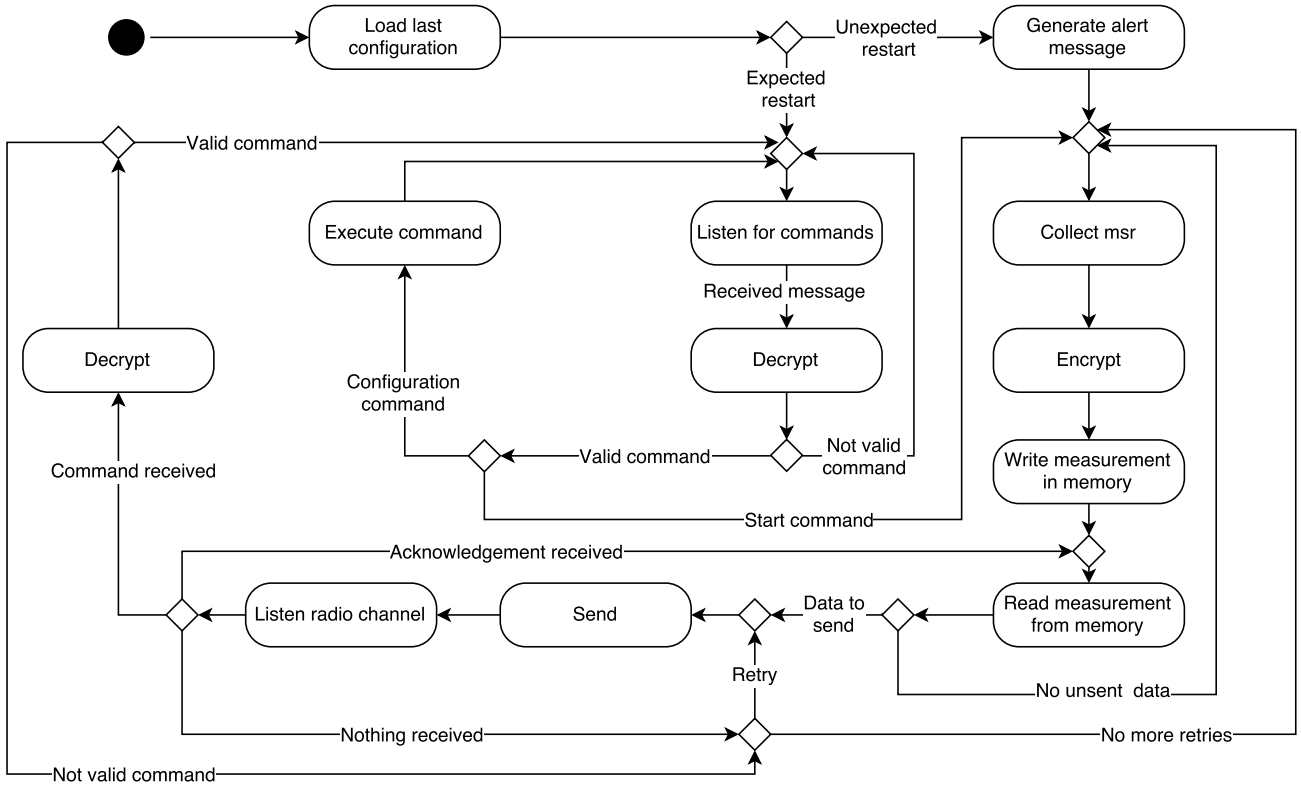

Figure 2: *Our designed software architecture of the metering device.*

recommended security strength for systems used after year 2030. Regarding ECC parameters our choice was the secp256r1 set with security level of 128 bits. For standard ECIES scheme we used the following algorithms:

Key agreement ECDH. According to the SEC 1 standard a choice can be made between ECDH and ECDH with cofactor included in shared secret computation. Since the latter was not directly supported in the ECC library we used, our choice was the general ECDH.

Symmetric encryption AES-128. According to NIST recommendation SP 800-57 AES is the only algorithm allowed having security strength of 128 bits.

Hash function SHA-256. It is one of currently widely used hash functions with our desired security strength. Key derivation function and MAC scheme in ECIES is built on a hash function.

Key derivation function ANS X9.63 function. Among the supported key derivation functions there is the ANS X9.63 function which is the only one with a described algorithm in the SEC 1 standard. Following this description we implemented the function.

MAC scheme Hashed MAC (HMAC) with SHA-256 and 256 bit key. Our hash function choice was SHA-256, which is also the basis of the MAC scheme.

In case of ECIES_ECDSA instead of a MAC scheme we used the ECDSA algorithm which outputs a 512 bit digital signature instead of a 256 bit MAC code. We will see in Section 5 that our evaluation includes both 128 bit and 256 bit security strengths, comparing the two ECIES schemes with another authentication technique (AES-GCM).

One of the major difficulties we faced with integration of ECC based methods was the need to split packets because of max packet size limitation of 66 B in the chosen wireless module, as described in Section 3.1. Actual not encrypted measurement size is only 10 B, however, with ECIES one measurement takes 112 B (64 B ephemeral public key, 16 B ciphertext and 32 B MAC code). Therefore, it needed to be split in two packets. One measurement encrypted and signed with ECIES_ECDSA method requires 144 B to be transmitted (64 B signature instead of 32 B MAC code). Hence, it needed to be split in 3 parts. Messages of measurements encrypted with AES-GCM didn't need to be split since the message size was 36 B, 16 B of them ciphertext, 16 B tag and 4 B GCM counter value.

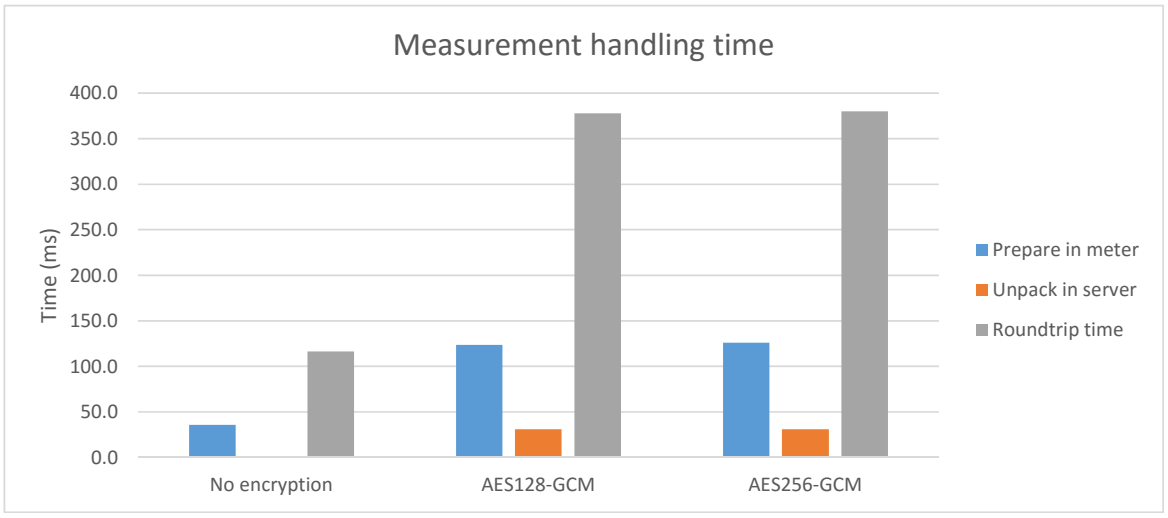

Figure 3: *Comparison of execution times for delivering measurement. Case of AES-GCM with 128 bit and 256 bit keys and no encryption.*

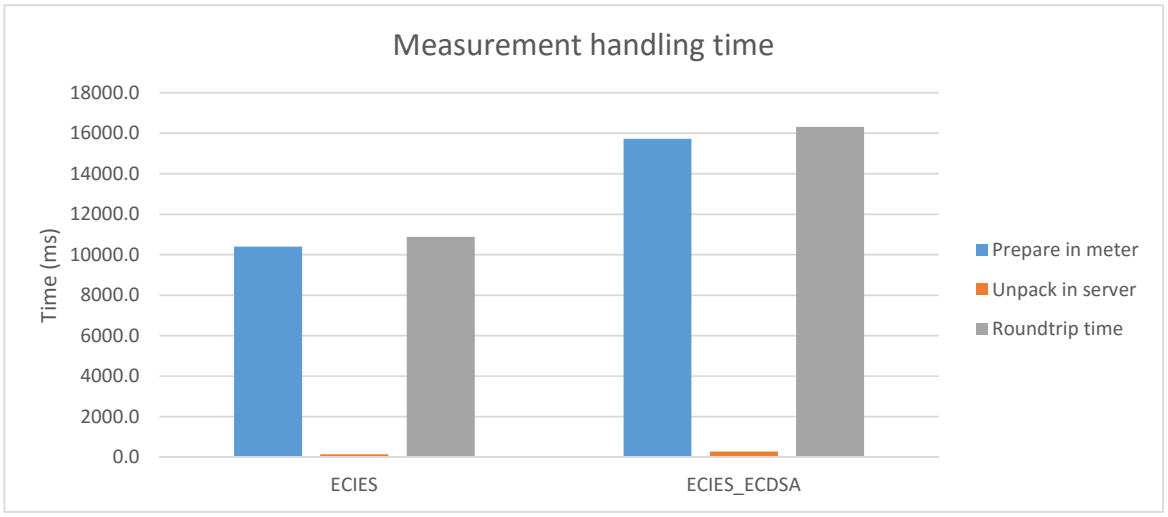

Figure 4: *Comparison of execution times for delivering measurement using ECIES and ECIES_ECDSA.*

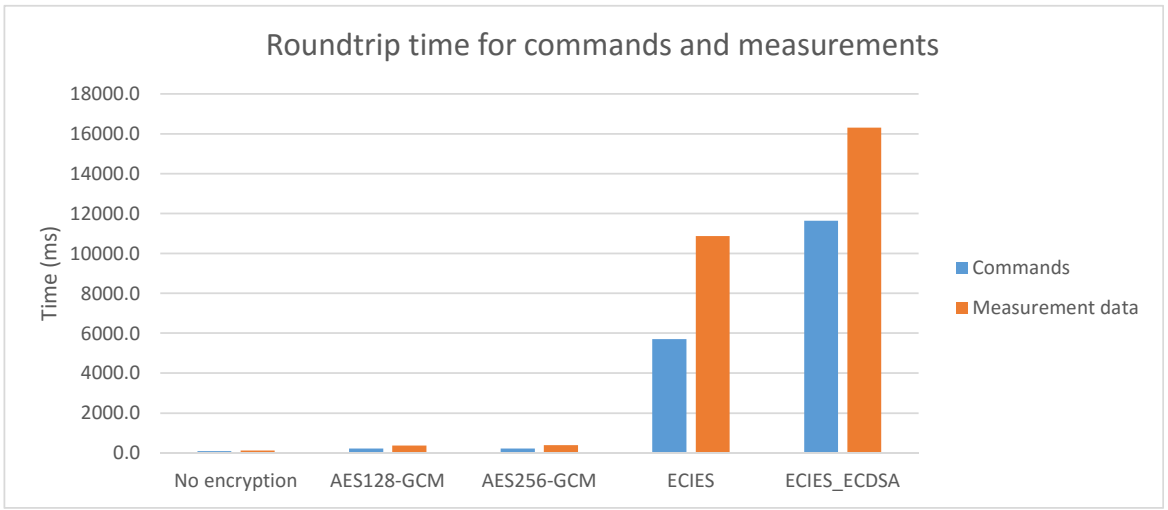

Figure 5: *Comparison of roundtrip times for command and measurement delivery.*

Note that the ECIES scheme relies on random key generation every time it is applied. In the MCU unused analogue ports can usually be used to get random noise. This, however, could not be achieved in our metering device since all the analogue input ports to the hardware were used. Pseudo random functions were used instead.

5. PERFORMANCE STUDY

The goal of our study was to evaluate the performance degradation for each cryptographic method.

5.1. Performance metrics

To evaluate how well the cryptographic methods perform on our platform, we chose 3 performance metrics.

1. Data preparing time - From raw data to ready-to-send packet. It includes the steps of encrypting data and writing the packet in memory.
2. Data unpacking time - From packet to raw data.
3. Roundtrip time - Includes the time to prepare data, send, unpack and receive acknowledgement of a successful transmission.

We chose these metrics to be able to compare the methods in detail. Data preparing and unpacking times give us a deeper insight into parts of roundtrip time. By looking separately at execution time on every device we can better understand how this system will perform when multiple smart meters are present.

5.2. Evaluation methodology

To evaluate the setup we performed 10 experiments. Measurement and command sending were considered as 2 separate experiments. Each of these are evaluated with ECIES, ECIES with ECDSA, AES-GCM in 128 bit and 256 bit versions, and no encryption respectively. Since there were small deviations in measurements, we did 100 measurements in every experiment and took the average value. We found the variation among the measurements were reasonably low, with a standard deviation ranging between 0.01 and 4.3 percent in the experiments.

5.3. Outcomes

In the case of measurement handling, Figure 3 shows that using AES-GCM encryption it takes roughly 3 times longer for a roundtrip compared to the case of no encryption. Furthermore, there is just 2 ms difference between using AES128 and AES256 bit versions. As expected, the difference between AES-GCM and ECC based methods, depicted in Figure 4 is huge. Even though time to decrypt a message in collector takes nearly 5 times longer in case of ECIES and 9 times longer for ECIES_ECDSA compared to AES-GCM, the bottleneck is, of course the meter. It requires around 82 and 125 times longer time for encryption operations using ECIES and ECIES_ECDSA respectively, compared to AES-GCM. Since in commercial smart metering systems data are usually sent every 30 minutes, 16 seconds for delivering measurements using a state-of-the-art asymmetric encryption scheme with digital signature can be considered as a good result.

Similar pattern as for measurement handling appears also for the case of commands. This is depicted in Figure 5. However, since it takes less ECC operations for decrypting messages than encrypting them the execution time is lower compared with the case of measurements. Note that in case of commands decryption takes place in the meter.

6. CONCLUSIONS

This work aimed at understanding the feasibility of encryption and authentication mechanisms in low-resource devices on open platforms in general. We explored the use case of a smart metering infrastructure, and designed a metering device with essential functions and various security building blocks to understand the trade-offs involved. The work points out that from a performance perspective it is feasible to build a smart meter using a low cost, low performance open source platform with asymmetric cryptography features. Performance loss in terms of execution time is much greater if ECC-based methods are used compared to using AES-GCM cryptography but for valuable assets it could be reasonable to choose higher security level instead of higher performance as long as frequency of meter readings is acceptable. Whether customer data, and the privacy/integrity requirements are considered to be valuable assets to dictate using high security standards is an issue that has wider social, political, and economic dimensions worthy of study. However, we posit that lack of security of commands sent to remotely operated sensors and actuators in a wide range of applications may have unforeseen implications for operation of critical infrastructures and regional/national assets.

One problem of our system was inability to get acceptable level of randomness because of hardware constraints. This is a problem worthy of further study. Furthermore, an important security aspect of smart metering devices is physical security

which we didn't consider at all. Future work includes studying how secure our system is against targeted attacks, including attacks in other parts of the infrastructure as well as the end devices.

ACKNOWLEDGEMENTS

This work was partially supported by RICS, the research centre on Resilient Information and Control Systems (www.rics.se), financed by the Swedish civil contingencies agency (MSB).

REFERENCES

Certicom Research (2009). Standards for efficient cryptography 1 (SEC 1), version 2.

Certicom Research (2010). Standards for efficient cryptography 2 (SEC 2), version 2.

Cleveland, F. M. (2008). Cyber security issues for advanced metering infrasttructure (AMI). In *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*.

Das, S., Y. Ohba, M. Kanda, D. Famolari, and S. K. Das (2012, August). A key management framework for ami networks in smart grid. *IEEE Communications Magazine 50*(8), 30–37.

Foudah, M. M., Z. M. Fadlullah1t, N. Katolt, R. Lu, and X. Shen (2011). Towards a light-weight message authentication mechanism tailored for smart grid communications. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*.

Good, T. and M. Benaissa (2008). *New Stream Cipher Designs: The eSTREAM Finalists*, Chapter ASIC Hardware Performance, pp. 267–293. Berlin, Heidelberg: Springer Berlin Heidelberg.

Gupta, V., M. Millard, S. Fung, Y. Zhu, N. Gura, H. Eberle, and S. C. Shantz (2005). Sizzle: A standards-based end-to-end security architecture for the embedded internet. In *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*.

Gura, N., A. Patel, A. Wander, H. Eberle, and S. C. Shantz (2004). Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *Cryptographic Hardware and Embedded Systems - CHES 2004 Volume 3156 of the series Lecture Notes in Computer Science*, pp. 119–132. Springer Berlin Heidelberg.

ICCS-NTUA,ÅF Mercados EMI (2015). Study on cost benefit analysis of smart metering systems in EU member states final report. Institute of Communication & Computer Systems of the National Technical University of Athens, ÅF Mercados EMI.

Kaplantzis, S. and Y. A. Sekercioglu (2012). Security and smart metering. In *European Wireless, 18th European Wireless Conference*.

Lake, D., R. Milito, M. Morrow, and R. Vargheese (2014). Internet of things: Architectural framework for ehealth security. *Journal of ICT Vol. 3 & 4*, 301–328.

Lang-Muhr, C., M. Schrattenholzer, and P. Tavolato (2015). Multi-layer agent-based simulation of network behaviour in advanced metering infrastructures. In *Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research 2015*.

Luan, W., J. Peng, M. Maras, J. Lo, and B. Harapnuk (2015, July). Smart meter data analytics for distribution network connectivity verification. *IEEE Transactions on Smart Grid 6*(4), 1964–1971.

Martinez, V. G., L. H. Encinas, and C. S. Avila (2010). A survey of the elliptic curve integrated encryption scheme. *Journal of computer science and engineering Volume 2, issue 2*.

Matt, B. J. (2005). The cost of protection measures in tactical networks. In *Proceedings for the Army Science Conference (24th), Orlando, Florida*.

McLaughlin, S., D. Podkuiko, and P. McDaniel (2010). Energy theft in the advanced metering infrastructure. In *Critical Information Infrastructures Security, Volume 6027 of the series Lecture Notes in Computer Science*, pp. 176–187. Springer Berlin Heidelberg.

Molina-Markham, A., G. Danezis, K. Fu, P. Shenoy, and D. Irwin (2012). Designing privacy-preserving smart meters with low-cost microcontrollers. In *Financial Cryptography and Data Security, Volume 7397 of the series Lecture Notes in Computer Science*, pp. 239–253. Springer Berlin Heidelberg.

National Institute of Standards and Technology (2016). Recommendation for key management - part 1: General (revision 4), SP 800-57.

Open (2016). Open energy monitor project. `www.openenergymonitor.org`, last visited January 2016.

Open Meter (2009). D1.1 report on the identification and specification of functional, technical, economical and general requirements of advanced multi-metering infrastructure, including security requirements. `www.openmeter.com`.

Raciti, M. and S. Nadjm-Tehrani (2013). *Critical Information Infrastructures Security: 7th International Workshop, CRITIS 2012, Lillehammer, Norway, September 17-18, 2012, Revised Selected Papers*, Chapter Embedded Cyber-Physical Anomaly Detection in Smart Meters, pp. 34–45. Berlin, Heidelberg: Springer Berlin Heidelberg.

The SecFutur project (2010). Deliverable 2.1, documentation of use cases, requirements and success factor indicators. `www.secfutur.eu/content/dam/sit/secfutur/en/publications/Deliverable_D2_1.pdf`.

Tudor, V., M. Almgren, and M. Papatriantafilou (2015). Harnessing the unknown in advanced metering infrastructure traffic. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, New York, NY, USA, pp. 2204–2211. ACM.