Zurich University
of Applied Sciences

School of
Engineering

InES Institute of
Embedded Systems

Crypto Acceleration for IoT

A Quantitative Comparison of Internal and External Solutions

Mario Noseda, Andreas Rüst

Zurich University
of Applied Sciences

www.zhaw.ch/engineering

Abstract – The exponential growth of IoT makes it an increasingly tempting target for attackers. In addition to pure software vulnerabilities (like weak or hardcoded passwords, insecure data transfer and storage), the OWASP IoT Top Ten[1] also lists hardware-related vulnerabilities such as the lack of physical hardening and the omission of a secure update mechanism. Mitigating these challenges requires additional hardware, as many of these devices are physically exposed and thus at a higher risk than conventional IT devices. More types of hardware solutions that implement the required functionality are entering the market; however, there are few to no performance comparisons. This lack further hinders the adoption of adequate solutions on a per-project basis. This white paper compares the performance of an MCU with an on-chip cryptographic engine to secure elements that connect to the host MCU over a serial bus.

Compared to conventional IT devices, IoT devices need to catch up in terms of security. The rapid increase of deployed IoT devices drastically increases the attack surface, affecting not only the IoT devices themselves, but also all related infrastructures, as the IoT devices might be used as an attack entry point. Energy-constrained devices are a major focus of attention, as various cryptographic algorithms cannot be used without significantly impacting power consumption.

Fortunately, more and more semiconductor manufacturers are addressing this problem. For several years now, secure elements have offered the capability of storing sensitive data (e.g., credentials and root certificates) in tamper-resistant memory and improving algorithms' energy and time efficiency through dedicated hardware. In addition, most of these secure elements are certified, which means that they are suitable for highly regulated devices. However, adding such a device does not come without its drawbacks. Even though the SPI/I2C communication can be encrypted, the lack of tamper-resistant memory in the MCU exposes the key on the host side to a determined attacker. Additionally, it is still possible to set up the communication completely unencrypted, although this is only intended for the development phase. Even though such a setting has an obvious right to exist, it could still lead to potential problems, e.g., forgetting to enable the encryption for the production build, similar to the well-known issue of forgetting to disable the debug ports.

In contrast, there are MCUs that have a cryptographic engine integrated as a peripheral. In direct comparison, they do not have the problem with the openly accessible interface, since the complete communication of the processor and the peripheral is handled via an internal bus. The on-chip solution also has a clear advantage regarding the data rate, since a significantly higher clock frequency can be used for internal buses. Although such MCUs usually do not contain tamper-resistant memory, the sensitive data is wrapped (encrypted and authenticated) with a special procedure within the crypto engine before being stored in standard flash memory. Compared to the tamper-resistant memory of secure elements, such security-focused MCUs offer a multiple of the storage capacity with a far superior wear-out characteristic.

There are a lot of other features (updateability, provisioning, certification, ...) that differentiate these two device classes. This white paper focuses on a performance comparison in terms of execution time and energy consumption.

For this, we designed a benchmark composed of various cryptographic primitives, conducted time and energy measurements, and then performed a statistical analysis of the gathered data.

[1] https://owasp.org/www-project-internet-of-things/

The benchmark is composed of the following cryptographic primitives:

- Generating random numbers (32 / 64 / 128 / 256 / 512 / 1024 bytes)
- Calculation of SHA256 hashes of the previously generated random numbers
- Generation of ECC (secp256r1) and RSA (1024 / 2048 bits) key pairs
- Calculation of ECDSA and RSA digital signatures
- Verification of ECDSA and RSA digital signatures

RSASSA-PKCS1-v1_5 was used as the RSA signature scheme for the complete project. For the remainder of the white paper, expressions like “RSA signature” in text, tables, or figures refer to this scheme.

As for the devices, we compared the Secure Crypto Engine 9 (SCE9) contained in the Renesas RA6M4[2] to the Infineon OPTIGA Trust M[3] and the NXP SE050[4] secure elements by measuring the respective execution time and energy consumption using a power analyzer during the benchmark.

Project Structure

We used the EK-RA6M4 evaluation kit for benchmarking the performance of the SCE9. The various cryptographic primitives were executed sequentially with an auxiliary GPIO set high during every test. The rising and falling edges allowed the exact calculation of the execution time and energy consumption during the post-processing of the power analyzer data.

Secure elements must be connected to a host MCU. To keep the benchmarks comparable, we kept the EK-RA6M4 and added our custom Secure Element Shield (SE-Shield), which contains, among other chips, the OPTIGA Trust M and the SE050. Again, we used the rising and falling edge of the auxiliary GPIO for processing the data.

The firmware has been written using the Renesas e^2 studio and their Flexible Software Package (FSP) version 3.1.0. Concerning the secure elements, we utilized the software development kit (SDK) provided by their respective manufacturer.

We applied basic optimization to the application (i.e., all unused clocks set to the lowest frequency possible and all unused peripherals turned off), but we were not able to optimize for both speed and energy consumption with a single setup. Therefore, clock speeds, prescalers, and the use of sleep modes were specifically configured for the operational aspect being evaluated.

Regarding optimization, the SCE9 is a fairly self-contained peripheral and only allows clock speeds to be adjusted, as shown in Table 1. The instruction clock of the core (ICLK) was set to the maximum of 200MHz when optimizing for execution time. In contrast, it was reduced to 100MHz when optimizing for energy consumption. In both cases, the PCLKA (clock source of the SCE9) was set to the maximum frequency of 100MHz. These frequencies yielded the best results for each corresponding metric. Furthermore, the “SCE9 Protected Mode” driver ensured that the SCE9 uses key protection measures equivalent to those of the secure elements by disallowing the support of plaintext keys and enabling simple power analysis (SPA) and differential power analysis (DPA) protections.

[2] https://www.renesas.com/ra6m4

[3] https://www.infineon.com/cms/en/product/security-smart-card-solutions/optiga-embedded-security-solutions/optiga-trust/optiga-trust-m-sls32aia/

[4] https://www.nxp.com/products/security-and-authentication/authentication/edgelock-se050-plug-trust-secure-element-family-enhanced-iot-security-with-maximum-flexibility:SE050

Table 1: Selected settings for optimizing the SCE9.

SCE9 Optimization	**Time**	**Energy**
Mode	SCE9 Protected Mode	SCE9 Protected Mode
PCLKA	100 MHz	100 MHz
ICLK	200 MHz	100 MHz

The optimization of the secure elements was determined empirically. We found that the I2C timing requirements could not be achieved for I2C fast-mode+ (1 MHz) without using extremely small pull-up resistors due to the length and capacitance of the traces. Since this only led to a marginal improvement in execution time and significant degradation in energy consumption, it was not used for either optimization. Thus, we used the I2C fast-mode (400 kHz) for both cases. Table 2 shows the configuration we determined for the secure elements. The ICLK was left at the maximum of 200 MHz for the time-optimized tests, since any reduction resulted in an increased execution time. The PCKLB (clock source for the I2C peripheral) was reduced to 12.5 MHz, as this was still a multiple of the I2C clock speed and did not lead to any performance degradation. In contrast, for energy optimization, the phase-lock-loop (PLL) was completely disabled and the ICLK was connected directly to the external crystal oscillator. This comparatively slow 24 MHz clock speed resulted in the best compromise between supply current and execution time, leading to the lowest energy consumption.

Furthermore, PCLKB could be halved again. Although the I2C peripheral can no longer reach the desired 400 kHz, this was the optimal operating point from an energy point of view. Lastly, both secure element SDKs contain various wait loops, in which the host MCU periodically (in the single-digit millisecond range) polls the secure element via I2C to determine whether or not the operation has been completed. If the MCU is put to sleep in between polling, additional energy can be conserved in exchange for a slight reduction in speed.

Table 2: Selected settings for optimizing the secure elements.

SE Optimization	**Time**	**Energy**
I2C Clock Speed	400 kHz	400 kHz
MCU Sleep	Never	Whenever possible
PCLKB	12.5 MHz	6 MHz
ICLK	200 MHz (PLL)	24MHz (XTAL / PLL off)

In terms of energy optimization, various settings have been determined so that the SCE9 as well as the secure elements consume as little energy as possible. However, optimizing the secure elements required a reduction of 88% from the maximum ICLK frequency and putting the MCU into a sleep mode whenever possible, which might not be feasible depending on your type of application. The reduction of the ICLK yields a significant performance penalty for the rest of the application. This could potentially be mitigated by dynamically switching the ICLK frequency before and after cryptographic calculations; however, this requires additional effort to maintain the functionality of all running peripherals (readjusting prescalers, etc.). Lastly, entering a sleep mode might not be possible due to high-availability constraints (e.g., hard real-time applications).

Measurement Setup

Figure 1 shows the setup used for measuring the performance of the three DUTs.

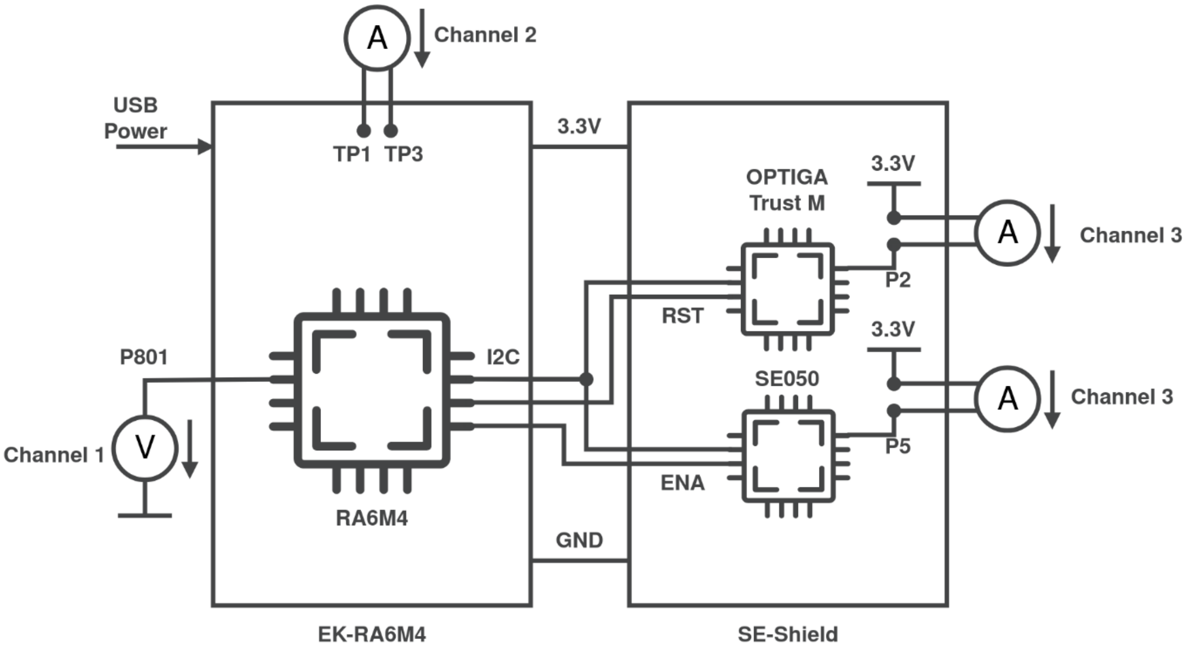


Figure 1: Measurement setup used for all benchmark measurements.

Channel 1 of the Keysight N6705B DC power analyzer was set up as a voltmeter and used to record the voltage of the P801 GPIO, which signaled the active execution of a benchmark test with a logic “1”. Channel 2 was set up as an ammeter and connected to the test pins TP1 and TP3 of the evaluation kit, which are intended to measure the supply current of the RA6M4 (only the MCU without anything else on the EK). For the OPTIGA Trust M and the SE050, channel 3 was set up as an ammeter and connected to either the P2 or P5 header on the SE-Shield for measuring the current consumption of the respective secure element.

A custom Python package allowed the automation of the measurements. It set up the power analyzer, started the measurement, reset the DUT, and finally exported the measurement results as a CSV file to a USB thumb drive plugged into the power analyzer. Afterwards, the package parsed all the CSV files and calculated all the conducted repetitions’ execution time and energy consumption. Finally, it exported the statistical analysis for all the tested cryptographic primitives.

Results

The following two statements are essential to prevent misunderstandings:

- The RSA key generation is a probabilistic operation (finding random large prime numbers is non-deterministic); therefore, the resulting measurements are only a general indication of the performance.
- The sampling period of the power analyzer was set to 81.92 µs (the fastest possible setting for this setup). Consequently, all median values approaching the sampling period must be regarded as neither accurate nor precise measurements, but rather as an approximation.

Absolute Performance

In order to measure the best possible values, the time-optimized setup was used for the execution time measurements and the energy-optimized setup for the energy consumption measurements. Table 3 and Table 4 list the resulting absolute values of the benchmark, where each value is the median of all measured repetitions.

As mentioned at the beginning of this chapter, values close to the sampling period of 81.92 µs could not be measured accurately. Consequently, all SCE9 measurements concerning the generation of random data, hashing, and the verification of an RSA signature are neither accurate nor precise. Therefore, they are marked accordingly.

Relative Performance

To better compare the performance of the SCE9 and the secure elements, Figure 2 shows the relative difference of the DUTs. It contains the ratios of the measurement results of the secure elements to those of the SCE9. For example, the OPTIGA Trust M takes 2.7 times longer than the SCE9 to generate a key pair on the secp256r1 curve. Similarly, the SE050 needs 0.729 times as much energy as the SCE9 to generate a 2048-bit RSA signature. For a quicker overview, the colors of the individual cells indicate the order of magnitude of the calculated ratios.

Table 3: Absolute execution time and energy consumption of the complete node during the random and hash benchmarks (median of values, rounded to 3 significant digits, n=(see chapter “Repetitions”))

		SCE9[1]		Trust M[2]		SE050[2]	
Primitive	Bytes	Time [s]	Energy [J]	Time [s]	Energy [J]	Time [s]	Energy [J]
Random	32	8.19e-05*	7.57e-06	7.13e-03	4.00e-04	1.84e-02	7.18e-04
	64	8.19e-05*	7.64e-06	7.78e-03	5.14e-04	1.93e-02	7.69e-04
	128	8.19e-05*	1.51e-05	9.01e-03	6.96e-04	2.05e-02	9.77e-04
	256	1.64e-04*	1.55e-05	1.34e-02	9.76e-04	2.87e-02	1.32e-03
	512	2.46e-04*	3.08e-05	3.09e-02	1.87e-03	3.91e-02	1.81e-03
	1024	3.28e-04*	4.68e-05	5.73e-02	3.75e-03	7.81e-02	3.62e-03
Hash	32	8.19e-05*	1.51e-05	3.92e-02	2.53e-03	2.14e-02	8.59e-04
	64	8.19e-05*	1.52e-05	4.60e-02	2.66e-03	2.24e-02	1.02e-03
	128	1.64e-04*	1.53e-05	4.73e-02	2.96e-03	2.65e-02	1.10e-03
	256	1.64e-04*	1.53e-05	5.72e-02	3.51e-03	3.78e-02	1.61e-03
	512	1.64e-04*	1.54e-05	7.00e-02	4.36e-03	5.11e-02	2.36e-03
	1024	1.64e-04*	1.56e-05	1.00e-01	6.23e-03	3.60e-01	1.44e-02

1. The energy consumption is calculated from the current consumption of the RA6M4
2. The energy consumption is calculated from the current consumption of the secure element and the host MCU (RA6M4).
*: Value is less than 20 sampling periods of the power analyzer. Consider this execution time as well as the corresponding energy consumption as approximations.

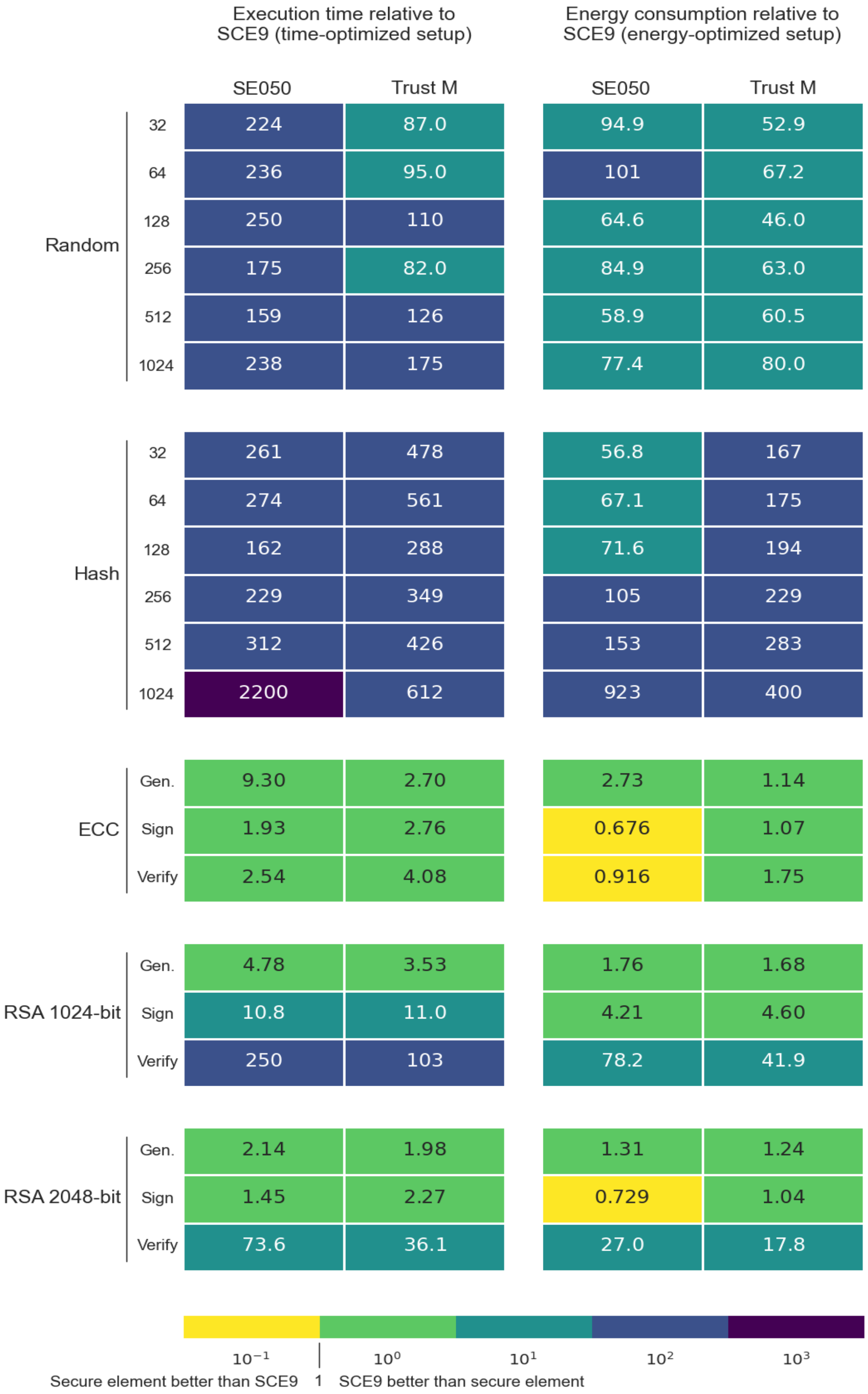
Table 4: Absolute execution time and energy consumption of the complete node during the asymmetric crypto benchmarks (median of values, rounded to 3 significant digits, n=(see chapter “Repetitions”))

		SCE9[1]		Trust M[2]		SE050[2]	
Primitive	Operation	Time	Energy	Time	Energy	Time	Energy
ECDSA	Key Gen.	2.38e-02	3.33e-03	6.44e-02	3.80e-03	2.22e-01	9.07e-03
	Sign	2.47e-02	3.44e-03	6.82e-02	3.69e-03	4.75e-02	2.33e-03
	Verify	1.87e-02	2.60e-03	7.61e-02	4.54e-03	4.75e-02	2.38e-03
RSA1024	Key Gen.	1.90e-01	2.51e-02	6.72e-01	4.23e-02	9.09e-01	4.44e-02
	Sign	6.31e-03	8.83e-04	6.94e-02	4.06e-03	6.79e-02	3.71e-03
	Verify	1.64e-04*	2.61e-05	1.69e-02	1.09e-03	4.09e-02	2.04e-03
RSA1024	Key Gen.	1.63e+00	1.87e-01	3.22e+00	2.32e-01	3.49e+00	2.46e-01
	Sign	1.31e-01	1.63e-02	2.97e-01	1.70e-02	1.90e-01	1.19e-02
	Verify	8.19e-04*	1.02e-04	2.96e-02	1.82e-03	6.03e-02	2.76e-03

[1] The energy consumption is calculated from the current consumption of the RA6M4

[2] The energy consumption is calculated from the current consumption of the secure element and the host MCU (RA6M4).

*: Value is less than 20 sampling periods of the power analyzer. Consider this execution time as well as the corresponding energy consumption as approximations.



		Execution time relative to SCE9 (time-optimized setup)		Energy consumption relative to SCE9 (energy-optimized setup)	
		SE050	Trust M	SE050	Trust M
Random	32	224	87.0	94.9	52.9
	64	236	95.0	101	67.2
	128	250	110	64.6	46.0
	256	175	82.0	84.9	63.0
	512	159	126	58.9	60.5
	1024	238	175	77.4	80.0
Hash	32	261	478	56.8	167
	64	274	561	67.1	175
	128	162	288	71.6	194
	256	229	349	105	229
	512	312	426	153	283
	1024	2200	612	923	400
ECC	Gen.	9.30	2.70	2.73	1.14
	Sign	1.93	2.76	0.676	1.07
	Verify	2.54	4.08	0.916	1.75
RSA 1024-bit	Gen.	4.78	3.53	1.76	1.68
	Sign	10.8	11.0	4.21	4.60
	Verify	250	103	78.2	41.9
RSA 2048-bit	Gen.	2.14	1.98	1.31	1.24
	Sign	1.45	2.27	0.729	1.04
	Verify	73.6	36.1	27.0	17.8

10^{-1}
1
10^{0}
10^{1}
10^{2}
10^{3}
Secure element better than SCE9
SCE9 better than secure element

Figure 2: Execution time and energy consumption of the secure elements relative to the SCE9 (ratio of median values, rounded to 3 significant digits, n=(see chapter “Repetitions”)).

Distribution

The highest spread, with a relative standard deviation (RSD)[5] of around 40 to 65%, occurred during the two RSA key generation tests, as expected from a non-deterministic algorithm. Furthermore, all tests in which the median is very close to the sampling period are bound to have a high RSD. Thus, we exclude these tests from our discussion on the distribution of the measured values, as this would arbitrarily increase the observed RSD.

Table 5 and Table 6 list the highest RSDs of the remaining tests for each DUT regarding the time and energy benchmarks, respectively.

Table 5: Execution time measurements with the highest RSD after excluding justified outliers.

SCE9[1,2]	**Trust M**[1]	**SE050**[1]
1.3% ECDSA Verify	12% RSA1024 Verify	0.2% Random 128 Bytes
0.63% RSA1024 Sign	6.8% Random 512 Bytes	0.19% Random 64 Bytes
0.13% ECC Key Gen.	4.0% Random 128 Bytes	0.17% Hash 64 Bytes

Table 6: Energy consumption measurements with the highest RSD after excluding justified outliers.

SCE9[1,2]	**Trust M**[1]	**SE050**[1]
1.4% ECDSA Verify	11% RSA1024 Verify	5.2% Random 64 Bytes
0.33% RSA1024 Sign	4.1% Hash 64 Bytes	0.35% Hash 1024 Bytes
0.11% ECDSA Sign	3.6% Random 128 Bytes	0.35% Random 32 Bytes

1. *Without RSA key generation.*
2. *Without random generation, hash calculation, and RSA verification.*

The SCE9 was too fast for the power analyzer in the random, hash, and RSA verify tests. Therefore, these tests had to be excluded from the RSD evaluation in addition to the RSA key gen. As a result, no clear statements can be made about the consistency of execution times in these tests. The remaining tests show a very high uniformity in execution time and energy consumption, with a maximum RSD of 1.3% and 1.4%, respectively.

Verifying a 1024-bit RSA signature with the Trust M resulted in the highest RSD overall with 12.4%. A closer look at the raw measurement data shows that most measurements took either 16.9 ms or 21.2 ms on average. Understandably, this also leads to a high RSD of the energy consumption during the same test. Unfortunately, we were not able to determine where this behavior comes from. Otherwise, the secure element also shows a very constant behavior with RSDs of 6.8% and lower.

The consistency of the execution time with the SE050 is surprisingly high at 0.2% RSD. It can be assumed that this is due to particularly stringent protection measures against timing attacks. Regarding the energy consumption, the generation of 64 bytes of random data resulted in significantly higher RSD compared to all other tests.

Experience from preceding projects shows that the random and hash benchmarks are less uniform regarding the observed metrics. Since these tests were removed from the SCE9 because the measurements were too imprecise, Table 7 and Table 8 now show the tests with the highest RSDs if the same tests are excluded for all DUTs to provide a fair comparison.

[5] RSD = 100 * σ / μ σ: standard deviation μ: arithmetic mean

Table 7: Execution time measurements with the highest RSD after excluding the same tests for all DUTs.

SCE9[1,2]	Trust M[1,2]	SE050[1,2]
1.3% ECDSA Verify	3.3% RSA1024 Sign	0.06% RSA1024 Sign
0.63% RSA1024 Sign	3.1% ECDSA Sign	0.02% RSA2048 Sign
0.13% ECC Key Gen.	3.1% ECDSA Verify	0.01% ECC Key Gen.

Table 8: Energy consumption with highest RSD after excluding the same tests for all DUTs.

SCE9[1,2]	Trust M[1,2]	SE050[1,2]
1.4% ECDSA Verify	2.3% RSA1024 Sign	0.26% RSA1024 Sign
0.33% RSA1024 Sign	2.0% ECDSA Sign	0.25% ECDSA Verify
0.11% ECDSA Sign	1.2% RSA2048 Sign	0.24% ECC Key Gen.

1. *Without RSA key generation.*
2. *Without random generation, hash calculation, and RSA verification.*

Ratio between RA6M4 and Secure Elements

Another observation we have made is the distribution of energy consumption between the secure elements and the host MCU RA6M4. Since secure elements always need to be connected to a host MCU, the selection of this MCU has a strong influence on the total energy consumption during the execution of one of the tested cryptographic primitives. Figure 3 shows the ratios with which the MCU and the secure elements contribute to the measured energy consumption.

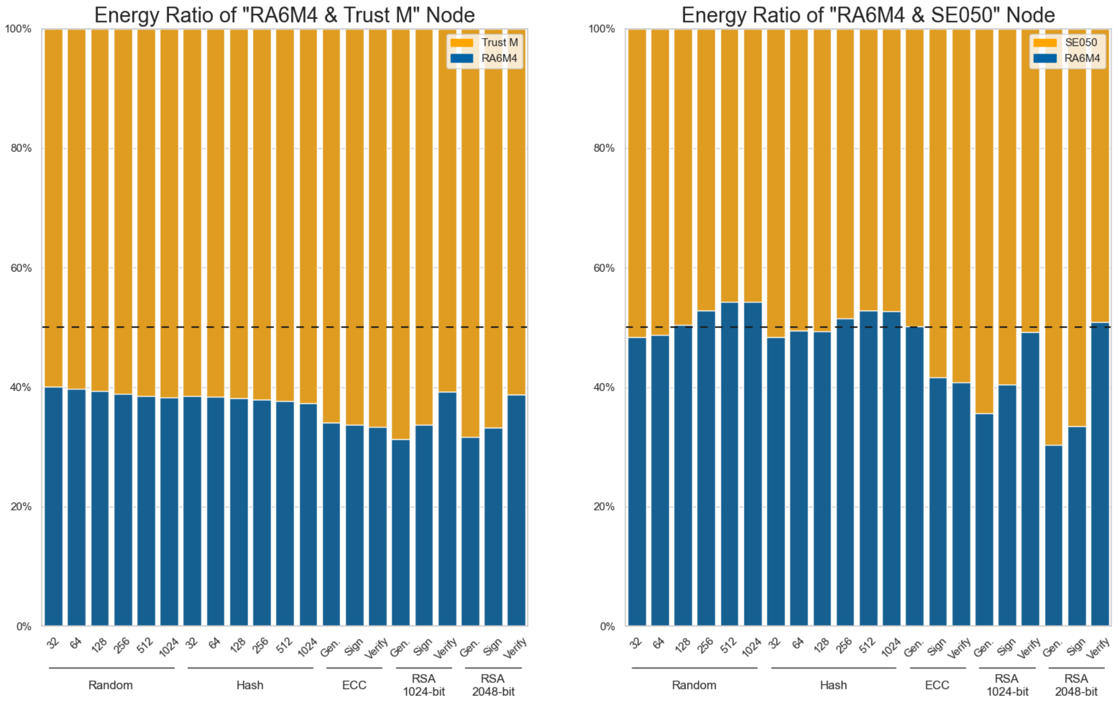


Figure 3: Energy ratio of the host MCU (RA6M4) and the secure elements while benchmarking the secure elements.

While the secure element executes the desired operation, the MCU waits and checks periodically (Trust M: 5 ms, SE050: 1 ms) whether the secure element has completed the operation. Thus, the sleep current and the time needed to enter and leave the desired sleep mode is significant when selecting the host MCU. If you want to use one of the two evaluated secure elements in a project with limited power availability, we recommend using a more energy-efficient MCU if possible, as the MCU should not account for a third to even more than half of the total energy consumption.

Repetitions

Table 9 contains the number of repetitions of the different tests. First, we recorded 100 time-optimized and 100 energy-optimized measurements with the SCE9. Unfortunately, the measurements with the power analyzer take a very long time despite automation (primarily due to exporting the data). Thus, the time-optimized measurements of the secure elements were stopped after 20 repetitions since a clear discrepancy in the execution time between the SCE9 and the secure elements could already be determined with only a few measurements. However, since the secure elements come significantly closer to the measured values of the SCE9 in terms of energy consumption, all 100 repetitions of the energy-optimized measurements were carried out.

Table 9: Number of repetitions performed for the various benchmarks.

n	Time-Optimized	Energy-Optimized
SCE9	100	100
Trust M	20	100
SE050	20	100

Conclusion

This white paper provides basic measurements of the SCE9 contained in the Renesas RA6M4 MCU, and the Infineon OPTIGA Trust M and NXP SE050 secure elements during the execution of cryptographic primitives. The results show that the different device types differ not only in their functionality and usage, but also in their performance in terms of execution time and energy consumption. Moreover, the devices used in this evaluation could not be optimized for both execution time and energy consumption simultaneously. Thus, the information contained in this document aims to support developers in finding an adequate solution for their given project by providing them with adequate measurement results.

Zurich University
of Applied Sciences

School of Engineering

The **Institute of Embedded Systems (InES)** is a leading research centre in the field of embedded systems with more than 50 employees as well as a state-of-the-art development and measurement infrastructure. Our services include:

- Design of system concepts and feasibility studies
- Consulting on the choice of technology
- Rapid prototyping and proof-of-concept
- Selection of radio components
- Realization of exhibition demonstrators
- Design of printed circuit boards (PCB)
- Writing microcontroller firmware: radio, protocols, sensors, actuators
- Interfacing to gateways and service platforms (cloud)
- Verification: Long-term tests and stress testing
- Embedded security: Threat analysis and protection measures

Mario Noseda, Andreas Rüst
Technikumstrasse 9
CH-8400 Winterthur

mario.noseda@zhaw.ch
andreas.ruest@zhaw.ch

www.zhaw.ch/ines

This research was funded by Renesas Electronics Europe GmbH.