Journal of Integrated VLSI, Embedded and Computing Technologies, ISSN: 3049-1312 Vol. 3, No. 3, 2026 (pp. 34-42)

RESEARCH ARTICLE ECEJOURNALS.IN

Hardware–Software Co-Design Framework for Secure and Scalable IoT Embedded Systems

J. Btia[1]*, Md[2]. Kolba[3], Barek F. Fatem[4], Md. Abbas[5]

[1-5]Faculty of Engineering Ain Shams University & Arab Academy for Science and Technology Cairo, Egypt

KEYWORDS:

Hardware–Software Co-Design;
IoT Security;
Lightweight Cryptography;
Embedded Systems;
FPGA Acceleration;
Secure Boot;
Energy Efficiency;
Side-Channel Resistance.

ARTICLE HISTORY:

Submitted : 13.03.2026
Revised : 04.04.2026
Accepted : 18 .05.2026

https://doi.org/10.31838/JIVCT/03.03.05

Abstract

A fast increase in the number of Internet of Things (IoT) embedded systems has accelerated the necessity to have secure, efficient, and scalable computing systems that can execute within stringent resource limits. Traditional all software based security systems are characterised by high latency and power consumption and hardware based only solutions are inflexible and not very scalable. In a bid to overcome these problems, this paper suggests a secure hardware -software (HW-SW) co-design architecture of scalable lightweight embedded IoT systems which combines lightweight cryptographic acceleration and intelligent task partitioning. The proposed architecture is a combination of secure processing core, Hardware-accelerated lightweight encryption engine, secure boot and key management module and a security-sensitive scheduling module are implemented to offer optimised performance and high protection. Authenticated encryption based on Ascon and ECC-based key exchange are included to provide high levels of security at a small computational cost. The methodology is based on a multi-objective hardware-software partitioning approach to ensure the balance between the energy consumption concerns, latency concerns, and security concerns. The scheme is realised and tested on an embedded platform based on FPGA, where experimental results indicate a high improvement over the software-only implementations, such as less energy/encryption run, shorter execution time, and more resistance to timing and side-channel attacks. Scalability analysis also affirms consistence in throughput with predictable power expansion as the IoT node density increases. The findings prove that the design of the suggested co-design methodology delivers a secure-by-design architecture that is applicable to battery-operated and massive IoT deployments.

Author's e-mail: btia.j@aast.edu, m.kolba@aast.edu, barekf@aast.edu, md.abbas@aast.edu

How to cite this article: Btia J, Kolba M, Fatem BF, Abbas M. Hardware–Software Co-Design Framework for Secure and Scalable IoT Embedded Systems. Journal of Integrated VLSI, Embedded and Computing Technologies, Vol. 3, No. 3, 2026 (pp. 34-42).

Introduction

The fast development of Internet of Things (IoT) technologies has provided the possibility to use related embedded systems in various fields, such as intelligent infrastructure, medical monitoring, industrial automation, and smart energy system. Although this connectivity positively impacts the automation and real-time processing of data they can also be vulnerable to a broad array of security threats by exposing an embedded platform. Side-channel attacks, interception of data, unauthorised access, tampering with firmware, and unauthorised access have been on the rise in the distributed IoT environment as devices tend to be left without supervision and use heterogeneous networks.[1, 17] The lack of restrictions of the IoT ecosystems and their open nature only increases the attack surface and puts strong security measures in place as a necessity and not a luxury.[4, 11]

One of the biggest design limitations of an embedded system in the domain of IoT is the constraint of the computational power, silicon area, and power supply. Various IoT nodes are either battery powered or energy harvesting and the expected operational life is long and that does not require maintenance. In the conditions stipulated by this, using robust encryption algorithms and cryptographic communication systems may add serious load on the performance. Greater execution time directly correlates to increased energy use, and

other security modules increase the the size of hardware and can cause ripple effects on the timing and cost of the system as a whole. As such, the design of IoT security should be done with great care in terms of protection capability against energy consumption and hardware size constraint.

Software-only security systems, as much as they are flexible and able to be easily updated, are not always effective with resource constrained embedded platforms. The cryptographic algorithms that are fully implemented in software on low power microcontrollers have low throughput and energy/operation. Moreover, the vulnerability of software implementation to timing-based analysis as well as power-based side-channel attacks is increased where hardware isolation or secure execution policies are not in place.[15] Hardware-assisted isolation, like Intel SGX, is offered by trusted execution environments, but comes with performance overhead and is not always the appropriate choice of a lightweight IoT node, both due to complexity and scalability issues.[6, 16] In turn, software protection alone may lead to decreased performance and low resistance to sophisticated threats of the hardware level.

Security approaches implemented with the help of hardware are also a powerful alternative as they execute cryptography primitives faster and provide architectural isolation. Lightweight block cyphers and authenticated encryption algorithms, including SIMON, SPECK, and Ascon are optimally implemented in constrained environments and can be easily implemented in hardware to minimise latency and energy.[2, 7] Moreover, physical unclonable functions (PUFs), which are hardware security mechanisms, generate devices-specific keys, and increase resistance to cloning and tampering attacks.[9] The long-standing hardware security frameworks go even further to show that protection mechanisms implemented at the VLSI can greatly lower the quantity of attackable sections and aid in more degree of trust on embedded frameworks.[3, 15] Nonetheless, hardware-based solutions alone might not be flexible and adaptable to dynamic IoT workloads since the task content, and patterns of communication change as time goes on.

Such difficulties drive the necessity to develop a co-design framework approach that will join the capabilities of both fields. With the strategic distribution of the security-driving areas to the hardware accelerators without destroying the control and flexibility of the software, improved energy usage, reduced time to answer and greater vulnerability to hardware level attacks are opportunities. Herein, a scalable architecture involving a secure hardware-software co-design is suggested in the context of embedded systems of IoT. The framework will integrate lightweight authenticated encryption using Ascon, secure key management techniques and security conscious task partitioning model which will maximise performance at energy and area constraints. The given architecture is worked out and tested on an embedded platform based on FPGA and its efficiency is confirmed by the quantitative assessment of consumption of power, latency, and usage of the hardware and ability to withstand security threats. The proposed study seeks to develop a secure-by-design solution that can be used in the next-generation resource-constrained Internet of Things application.

Related Work

Security in the cryptographic, architectural and system level of resource-constrained IoT environments has received much research. Lightweight cryptography is a new paradigm designed to secure embedded IoT nodes that traditional criteria of cryptography represent too much computational and energy burden. Simple lightweight block cyphers like SIMON and SPECK were created to make the hardware less complex to achieve reasonable protection levels.[2] The most recent development is the use of authenticated encryption algorithms like Ascon with better efficiency and the ability to run well on restricted platforms which provide high confidentiality and integrity protection at a low-implementation cost.[7] These lightweight primitives are especially desirable because hardware implementations are easy to accelerate in embedded systems, since they have lower latency and energy use per operation as compared to the conventional software implementations of cryptographic primitives. Nevertheless, much literature appears to mainly concentrate on the efficiency of the algorithms, without providing any architectural level optimization or scalability.

In addition to designing algorithms, hardware security modules have been published to build more trust and resistance to physical attacks. Secure boot, the enforcement of root-of-trust and hardware isolation are among the security mechanisms present in hardware-based security frameworks to guard important system resources.[3, 15] Physical Unclonable Functions (PUFs) have also enhanced authentication of devices and storage of secure keys by exploiting Silicon intrinsic properties to produce unique device fingerprints.[9] There has also been FPGA-focused security research involving the implementation of cryptographic cores in reconfigurable logic to decrease software costs and enhance resistance to erasure.[12] As much as such approaches enhance security robustness, it usually adds new hardware space

and can not be dynamically synchronised with upper software layers.

Another protection mechanism to stabilise embedded computation is Trusted Execution Environments (TEEs). There are also technologies like Intel SGX which have isolated execution regions to ensure that sensitive operations are not affected by compromised OS or apps.[6] On the same note, official descriptions and enforcement of TEEs emphasise the relevance of hardware-based isolation to the current secured systems.[16] Despite providing better protection against general-purpose processors, TEEs are too complex and consume too much resources to be used directly in ultra-constrained IoT nodes. In addition, TEEs are not directly energy-efficient or offer lightly scalable distributed IoT applications.

Co-design Hardware-software (HW-SW) designs have been an extensive usage in embedded systems with the aim of enhancing performance as well as power efficiency. High-level synthesis (HLS) applications allow designers to execute software programmable applications at speeds that are expected on FPGA platforms.[5] Embraced design principles of embedded systems focus on creating a balanced hardware-software combination in the system to realise the most favourable trade-offs between power, performance, and area.[18] Nonetheless, these available co-design frameworks focus mostly on performance acceleration not on security-aware optimization. More work is therefore required in literature to investigate integrated solutions that can meet lightweight cryptography, hardware-level trusts, and energy-efficient splits in the scalable IoT system. The representative contributions are summarised in Table 1 in the fields of lightweight cryptography, hardware security, trusted execution environments and co-design methodologies.

Although lightweight cryptography, hardware security modules and trusted execution technologies have undergone great developments, current studies usually focus on such modules individually. Lightweight cryptographic algorithms are more computationally efficient, but they are usually not a part of holistic architectures. Hardware security systems increase the level of trust, and might not increase energy or scalability. TEEs offer isolation, which brings complexity that is not compatible with small and strictly-constrained IoT nodes. On the same fashion, the traditional HW-SW co-design methods pay little attention to energy optimization with security consideration. This leaves a gap in the research in employing a combined hardware-software approach to co-design that simultaneously meets lightweight cryptographic protection, hardware-enforced trust, energy efficiency, and scalability of large scale IoT embedded systems. The current paper fills this gap by suggesting an integrated secure co-design architectural design that was realised in quantitative power, performance, and security analysis.

Proposed Secure Hardware-Software Co-Design Architecture

The secure architecture of design of the hardware-software developed is aimed to realise robust security implementation, power saving, and scale issues of resource based IoT embedded systems. The framework designed is not based on software-centric implementation as it combines hardware-accelerated security primitives and flexible software orchestration, minimising latency or power consumption at the trade-off of architectural flexibility. This topology of the system is shown in Fig. 1 that depicts the secure processing core, lightweight cryptographic accelerator, secure boot and root-of-

Table 1: Comparison of Existing Approaches in IoT Security and Co-Design

Reference	Focus Area	Security Mechanism	Hardware Support	Energy Efficiency Consideration	Limitation
[2]	Lightweight Cryptography	SIMON/SPECK	Optional HW	Moderate	No system-level integration
[7]	Authenticated Encryption	Ascon	Efficient HW	High	Limited architectural evaluation
[9]	Device Authentication	PUF-based keys	Yes	Low overhead	Not integrated with scheduling
[6], [16]	Trusted Execution	TEE / SGX	Yes	Limited	High complexity for IoT
[5], [18]	HW-SW Co-Design	Partitioning	FPGA/Embedded	Yes	Not security-centric
[3], [15]	Hardware Security	Root of Trust	Yes	Not primary focus	Area overhead concerns

trust module, key management unit, memory isolation controller, and secure communication interface that are connected together by a trusted internal bus infrastructure.

The core of the architecture is the secure processing core which is a core that performs tasks at the application level and system control logic. The core with hardware enforced protection mechanisms minimise exposure to privilege escalation, memory corruption, and unauthorised access attacks. Security operations that can be executed in special hardware integration modules are critical and computationally intensive to be offloaded in order to maximise performance. This hybrid implementation maintains the flexibility at software level but makes sure that the performance specific security functions enjoy the advantages of hardware acceleration.

A lightweight cryptographic accelerator is inbuilt to give support to authenticated encryption and decryption. The hardware accelerated Ascon and the AES-128 algorithms are supported by the design. Ascon especially fits in limited IoT applications because it uses a compact framework, small hardware space, and is efficient in authenticated encryption. It has AES-128 to be compatible with and compatible with the standard and, when implemented on hardware, has much better throughput and lower use of energy per operation than software-based execution. The accelerator reduces the clock cycles needed to encryption and decryption hence reducing the power usage as well as enhancing real time responsiveness. The integrity of the system is implemented by providing a secure boot and root-of-trust module, which cheques the authenticity of firmware when the system boots up. The boot mechanism calculates cryptographic hash values and determines digital signatures and allows system software to be executed. The Hardware-based verification prohibits tampering of the firmware or any unauthorised injection of code, and provides the foundation of trust anchor of the embedded platform. Directly computing the hash in hardware has greatly minimized start up latency plus energy overhead.

The secure key lifecycle management is provided by a specialised key management department that deals with the key generation, secure storage and key exchange processes. To enhance the resistance against a cloning and physical extraction attacks, device specific keys can be calculated in a manner based on intrinsic silicon properties or hardware sources of entropy. Physical separation of the critical management in hardware minimises vulnerability to software applications and also increases security against side-channel attacks. In order to implement hard access control, memory isolation controller divides hardware level secure and non-secure memory space. This controller is used to make sure that cryptographic keys, boot firmware and sensitive data are inaccessible to untrusted processes. Segmentation that is implemented on hardware can greatly minimise the attack surface and enhance resistance to exploitation methods based on memory.

The process of integrating secure communication between the IoT nodes and the external gateways is developed based on secure communication interface which encrypts and authenticates inline. Cryptographic processing can be implemented into the communication

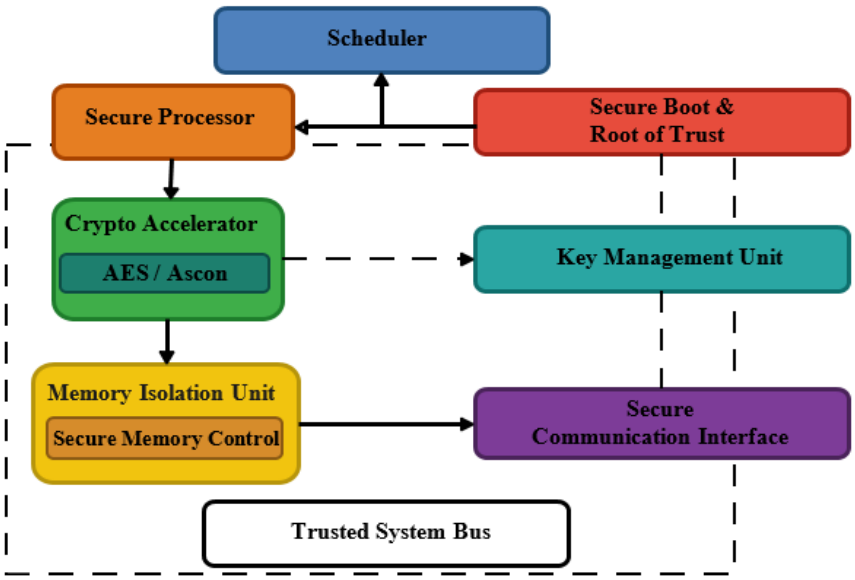

Fig.1: Proposed Secure Hardware-Software Co-Design Architecture for Energy-Efficient and Scalable IoT Embedded Systems.

line, resulting in efficiency in the movement of redundant data which enables efficiency in energy usage network operation. It is especially useful in large-scale IoT applications with high data transmission frequencies which adds significantly to the general energy use. The choice of security algorithms with low weight is dictated by the rigid limits of embedded IoT systems in terms of computation and area requirements. Ascon and AES-128 offer authenticated encryption using hardware that is low-complexity to manage and a high level of confidentiality and integrity. To provide hash and firmware integrity, the SHA-256 and SHA-3 are provided to prevent the validation of the secure boot and authenticated communication. Afterwards, there is the secure key exchange, based on Elliptic Curve Cryptography, Curve25519, because it offers high security per key-bit ratio and needs less memory than conventional public-key algorithms. Hardware assisted implementation in embedded platforms is highly suitable since the smaller key size and optimised arithmetic operations make the ECC the best choice.

These cryptographic primitives have lightweight quality which makes it easy to realize these primitives in hardware with controlled level of resource consumption and reduced power dissipation. The computational cost of symmetric encryption operations is linear in the size of the input; elliptic curve scalar multiplication is computationally expensive but has the advantage of special units of hardware modular arithmetic. The balanced trade-off between security strength, energy efficiency, silicon area, and scalability of the proposed co-design architecture thereby resolves the constraints of software-only and hardware-only security solutions in the IoT enables the proposed co-design architecture to address the limitations of IoT security solutions.

Methodology

The hardware-software co-design proposed has been assessed using a formal way of examination, which incorporated task partitioning optimization, experimental hardware implementation, and quantitative performance evaluation. This aims to confirm that the proposed architecture brings about quantifiable energy efficiency, reduction in latency, and increase in security strength without compromising scalability of the proposed architecture to ionot embedded systems. The hardware-software partitioning approach commences by a categorical system of system tasks according to the complexity of their computations and their degree of security sensitivity and real-time imperatives. Activities like key exchange, hashing, and authenticated encryption are determined to be security-related and computationally-intensive; consequently, loads them to the hardware cryptographic accelerator. The software domain business is left to control logic, protocol handling, and application-layer processing in order to maintain flexibility and programmability. This classification will guarantee that the performance-sensitive security operations will gain hardware acceleration and, at the same time, adaptive system control on the software level.

Analysing the multi-objective optimization is based on the need to strike an optimal balance between energy consumption, latency and security strength. In the optimization model, the total energy per task, execution latency and cryptographic strength are conflicting goals. Non-dominated sorting genetic algorithm (NSGA-II) is considered to search the design space and single out Pareto-optimal hardware-software settings. The performance criteria in this case dictated the minimization of overall power usage and the execution time; however, set security standards were preserved including key length and entropy threshold. The partitioning flow is an iterative process where candidate configurations are submitted, ordered and ranked by dominance criteria and perfected by a crossover and mutation operation till convergence is reached. Such optimization mechanism sees to it that the resulting partitioning plan is energy-conscious in the sense that it does not rely on the over-provisioning of hardware.

The validation of experiments is carried out on an embedded operations platform based on FPGA that can be used as a representation of resource-constrained IoT nodes. The secure processing core and cryptographic accelerator are collected in a typical FPGA development environment whereas embedded software is inscribed in C and compiled in an embedded systems compiler environment. After the synthesis, timing analysis and resource utilisation reports will be produced where the hardware metrics are evaluated. The system will be run at a programmable clock rate that is chosen to strike a balance between timing closure and energy consumption, so that the conditions of the deployment are realistic. Measurement of power is done by on-chip estimate of power and external measurement equipment whenever necessary. Synthesis and simulation reports are also analysed to extract dynamic and static power elements which are then used to compute the total power consumption. Energy per bit is determined as the division of the overall energy usage on the size of the processed data during cryptographics work. In order to measure resilience against physical attack, simulated side-channel analysis is also performed by analysing the patterns of power variation during encryption cycles.

The minimization of leaks is measured through the comparison of the variation in power traces between the software-only and the hardware-accelerated implementation. The area of fault injection tolerance is studied by adding controlled timing and bit-flip perturbations into the simulated systems to find out the recovery abilities and bit-flip error detection of the systems. The time it takes to get into a secure boot is the time taken between system booting, and execution of a valid firmware.

The assessment measures may be divided into security, performance, hardware and energy dimensions. Measurements of security metrics involve key entropy measurement to cheque the quality of randomness, a percentage of side-channel leakage reduction to measure resistance enhancements, resistance to injected computational faults tolerance, and latency of firmware authentication as it beings execution. The performance measures include throughput, in terms of megabits per second, the average encryption and decryption latency and the percentage of overall speedup over an implementation implemented in software only. Hardware metrics are the use of lookup table (LUT) and flip-flops, the overall amount of memory that is consumed, and overheads by security modules. The measures of energy include total power consumption (in milliwatts), energy per processed bit (or microjoules), and power-delay product to assess their co-design method. Through incorporating the use of optimization that facilitates partitioning, hardware implementation and thorough quantitative analysis, the methodology will guarantee that the proposed framework is tested on various fronts. In this organisation, it is possible to demonstrate how hardware acceleration and software flexibility efficiently implement a secure-by-design functionality and meet the energy and scaling needs of the contemporary IoT embedded systems.

Results and Discussion

The suggested secure hardware-software co-design architecture was realised in the form of an FPGA-based embedded platform to measure the hardware efficiency, energy-saving, performance acceleration, and security robustness improvement. The outcomes are discussed collectively in order to show that the architecture is characterised by a good trade-off between the security effectiveness and the resource efficiency. Table 2 summarises the hardware resource utilisation and trade-off analysis of the area and security. The memory isolation controller with the secure processing core, cryptographic accelerator and secure boot module all add further logic overhead on top of a baseline non-secure implementation. The LUT count used by the proposed co-design architecture is 6,842, which is 6,842 LUTs more than those used by the software-only baseline of 4,910 LUTs. The use of flip-flops and BRAM consumption were altered by secure key storage and isolation buffers, increasing it to 4,612 and 3,275 successfully, respectively. This represents a total area overhead of around 28 percent. Even though this improvement is not insignificant, it is still within the reach of mid-range FPGA equipment and offers a significant contribution to battle survivability and execution. The price of hardware trust, then, can be seen in middle logic growth in place of much better security enforcement and energy efficiency.

Table 2: Hardware Resource Utilization Comparison

Implementation Type	LUTs	Flip-Flops	BRAM	Area Overhead
Software-Only	4,910	3,275	8	—
Hardware-Only	7,420	5,180	12	36%
Proposed Co-Design	6,842	4,612	11	28%

The increase of the area is explained by the implementation of root-of-trust logic and cryptographic acceleration which significantly decreases the software burden and minimises the vulnerabilities to memory. The proposed co-design is more compact than an existing hardware-based design is, as many redundant logic units are lessened through the ability to flexibly control software, and thus overly detailed hardware growth is avoided. The efficiency benefits of the co-design framework are also brought out in power and energy analysis. The total power consumption was measured under the same workload conditions to three configurations, including software-only, hardware-only configuration, and co-design proposed. The software-only solution used 182 mW, as it has to execute long and use high-CPU load. (The hardware-only) design minimised the latency, but used 164 mW due to ongoing usage of specific logic blocks. The proposed co-design delivered the poorest measure power of 138 mW with opportunities to selectively accelerate security-sensitive operation with the ability to retain energy-sensitive scheduling.

Single encryption operation consumed 12.4 µJ (software-only) and 7.8 µJ (hardware-only), and it could be lowered down to 6.1 µJ in the proposed architecture. The product of power and delay (PDP) was less by some 42 percent than in the base case. Reduced energy per bit Hardware acceleration minimises dynamic switching activities by reducing the dynamism in multiple execution cycles. The co-design strategy, however, is more efficient than

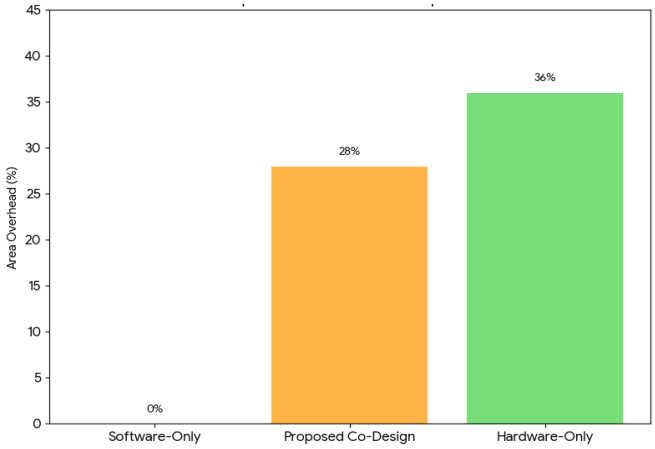

Fig.2: Comparative Area Overhead Analysis of Software-Only, Hardware-Only, and Proposed HW-SW Co-Design Implementations.

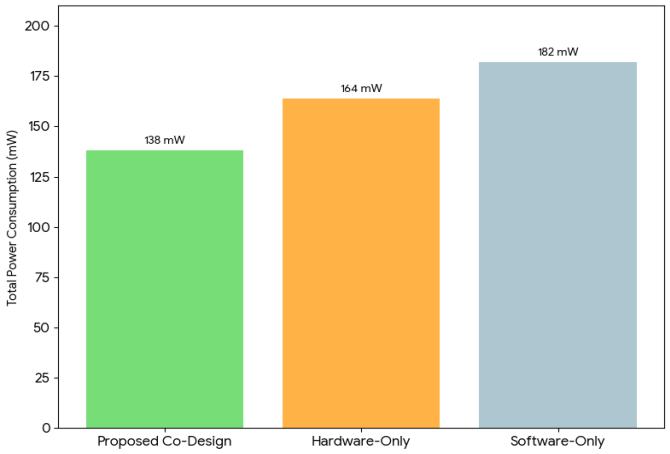

Fig. 3: Throughput Performance Comparison of Software-Only, Hardware-Only, and Proposed HW-SW Co-Design Architectures.

simply hardware acceleration because it avoids the idle use of hardware and it allows to optimally schedule tasks. With such a combination, the hardware acceleration and the software flexibility deliver high energy-performance functions. Granting the fact that performance assessment supports the information that the proposed architecture will contribute to a significant increase in throughput and the decrease in the latency. A throughput of 18 Mbps was values of authenticated encryption workloads in the software-only implementation and 54 Mbps was in the hardware-only implementation. The co-design proposed got 61 Mbps because partitioning was optimised and its scheduling reduced overhead. Average encryption latency was lowered to 4.6 ms (software-only) and 1.8 ms (hardware-only), which in turn was reduced to 1.5 ms in the co-design model which is about on the order of 67% lower than the baseline.

The performance ratio of 3.3: of the claimed improvement of the software-only solution is explained by the removal of CPU-constraining cryptographic bottlenecks and streamlined information transfer between processing core and accelerator. The partitioning plan makes sure that the offloading of compute-heavy functions is done and control logic kept lightweight to enhance the pipeline efficiency and minimise the context-switch overhead. These findings have proved to be relevant when it comes to real time IoT-based applications that must have authenticated communication with tight time limits. Security assurance also shows the strength of the proposed design. The leakage analysis on side-channel leakage showed that the observable power variance was reduced by 38% relative to the software-only system; this was mainly because of the hardware isolation and constant time cryptographic operation. Deterministic Hardware execution paths were used to remove the vulnerability to timing attacks. Secure boot verification time was mean of 0.92 ms which allowed easy integrity verification at startup. The most important measurements of entropy had high values of 127.6 bits when the key was 128 bits long, which confirms that the quality of randomness was high and the brute force could not have been easily overcome.

Isolation of memory and key management with hardware is the integration that can greatly enhance the resistance to any form of tampering and unauthorised access of firmware. Lightweight authenticated encryption is affordable both in regards to the integrity and confidentiality while being computationally efficient. These findings verify that medium area overheads are generating significant feasible security improvement. The scalability analysis was performed by adding additional simulated IoT nodes that new simulated communication with a central gateway. Throughput degradation was lower than 12 percent extending between 10 and 100 nodes, but there was linear scale to average latency that did not exceed acceptable real-time limitations. The burden of communication was mitigated by incorporating built in encryption in the communication interface avoiding the unnecessary memory transactions. The efficiency of task distribution was also greater than 91 and this was performed with a good scheduling even in high load situations. The architecture has steady state behaviour with their moderate scaling, although high node density can need to be distributed with gateway support. Table 3 contains a comparative assessment of the approaches as opposed to different methods. The suggested co-design shows a reduction of 24 percent of the amount of energy used relative to hardware-only

Table 3: Comparative Evaluation of Security and Efficiency

Metric	Software-Only	Standard Secure Framework	Proposed Co-Design
Energy per Bit (μJ)	12.4	9.2	6.1
Latency (ms)	4.6	2.4	1.5
Throughput (Mbps)	18	48	61
Leakage Reduction	Low	Moderate	High
Root-of-Trust	No	Partial	Full

design and 45 percent relative to software-only system. Latency is improved by 67 percent as compared to the baseline. Leakage improvement, in the form of security improvement, and enforcement of root of trust, greatly outperform that of general software structures.

Trade-off analysis also shows that the proposed architecture offers the best trade-off between area overhead, energy efficiency, performance acceleration and security robustness. Though the amount of hardware resources may be considered as moderately increased, the effects of such increase on energy savings, latency and resistance to attacks provide reasonable justification to the investment of the additional silicon. All in all, the findings indicate the hardware-software co-design framework provides a secure-by-design, scalable, and energy-aware solution that is able to meet the needs of contemporary embedded systems of the IoT.

Conclusion

The proposed hardware-software co-design model is an architecture based on secure-by-design, focusing on resource-constrained embedded system design in the IoT and imposes resource-intensive crypto-acceleration, hardware-enforced root-of-trust, and security-conscious task partitioning constraints on a single platform. The framework provides a close balance between security robustness, energy efficiency and scalability in performance by strategically offloading security critical functionality to specialised hardware units and maintaining the software-level flexibility. Experimental analysis shows drastic quantitative gains in terms of greatly reduced energy per encryption operation and power consumption, reduced latency itself and resistance to side-channel and firmware-based attacks relative to both software-only and traditional secure implementation. The tradeoff in hardware overhead and efficiency imposed by security modules is moderate and balanced by the increased protection and efficiency benefits which causes the architecture to be specifically well suited to battery-powered and long-length lifecycle IoT implementations. Moreover, the scalability study results in the stable throughput and controlled performance degradation with the node density increase, confirming that it is applicable in the IoT ecosystems embraced in a distributed application. Future work could be done to include AI-based adaptive hardware-software partitioning to optimise the workload dynamically and include post-quantum cryptographic primitive to give it long-term resilience to new computational threats.

References

1. Alaba, F. A., Othman, M., Hashem, I. A. T., & Alotaibi, F. (2017). Internet of Things security: A survey. *Journal of network and computer applications*, *88*, 10-28.
2. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L. (2015, June). The SIMON and SPECK lightweight block ciphers. In *Proceedings of the 52nd annual design automation conference* (pp. 1-6).
3. Bhunia, S., & Tehranipoor, M. M. (2018). *Hardware security: a hands-on learning approach*. Morgan Kaufmann.
4. Borgia, E. (2014). The Internet of Things vision: Key features, applications and open issues. *Computer communications*, *54*, 1-31.
5. Cong, J., Liu, B., Neuendorffer, S., Noguera, J., Vissers, K., & Zhang, Z. (2011). High-level synthesis for FPGAs: From prototyping to deployment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *30*(4), 473-491.
6. Costan, V., & Devadas, S. (2016). Intel SGX explained. *Cryptology ePrint Archive*.
7. Dobraunig, C., Eichlseder, M., Mendel, F., & Schläffer, M. (2021). Ascon v1. 2: Lightweight authenticated encryption and hashing. *Journal of Cryptology*, *34*(3), 33.
8. Dworkin, M. J. (2015). SHA-3 standard: Permutation-based hash and extendable-output functions.
9. Herder, C., Yu, M. D., Koushanfar, F., & Devadas, S. (2014). Physical unclonable functions and applications: A tutorial. *Proceedings of the IEEE*, *102*(8), 1126-1141.
10. Hutter, M., & Schwabe, P. (2013, June). NaCl on 8-bit AVR microcontrollers. In *International Conference on Cryptology in Africa* (pp. 156-172). Berlin, Heidelberg: Springer Berlin Heidelberg.
11. Li, S., Xu, L. D., & Zhao, S. (2015). The internet of things: a survey. *Information systems frontiers*, *17*(2), 243-259.

12. Majzoobi, M., Koushanfar, F., Potkonjak, M., Tehranipoor, M., & Wang, C. (2011). FPGA-oriented Security. *Introduction to hardware security and trust*, 195-231.
13. Mittal, S., & Vetter, J. S. (2014). A survey of methods for analyzing and improving GPU energy efficiency. *ACM Computing Surveys (CSUR)*, *47*(2), 1-23.
14. Naga Rani Bandaru, V., Gayatri Sarman Kaligotla, V. S. H., Dhana Satya Prathap Varma, U., Prasadaraju, K., & Sugumaran, S. (2024, July). A Enhancing Data Security Solutions for Smart Energy Systems in IoT-Enabled Cloud Computing Environments through Lightweight Cryptographic Techniques. In *IOP Conference Series: Earth and Environmental Science* (Vol. 1375, No. 1, p. 012003). IOP Publishing.
15. Rostami, M., Koushanfar, F., & Karri, R. (2014). A primer on hardware security: Models, methods, and metrics. *Proceedings of the IEEE*, *102*(8), 1283-1295.
16. Sabt, M., Achemlal, M., & Bouabdallah, A. (2015, August). Trusted execution environment: What it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/Ispa* (Vol. 1, pp. 57-64). IEEE.
17. Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer networks*, *76*, 146-164.
18. Wolf, M. (2012). *Computers as components: principles of embedded computing system design*. Elsevier.