Dynamic Inertial Sensing for Device Identification

Michal Borowski[a], Gareth Howells[b]
School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK
m.borowski@essex.ac.uk, gareth.howells@essex.ac.uk

Keywords: ICMetrics, IoT security, Machine learning, Key generation

Abstract: Previous studies have demonstrated that inherent imperfections in inertial sensors can serve as unique identifiers for devices, supporting secure key generation and device authentication. However, these works have primarily focused on measurements taken under static conditions, where sensors were at rest. This paper extends that line of research by investigating inertial sensor-based device identification under dynamic conditions. Specifically, when devices are in motion within a vehicle. We demonstrate that despite motion-induced variations, consistent inter-sensor relationships can be exploited to accurately identify individual devices. Furthermore, we present an ICMetric key generation approach that leverages accelerometer and gyroscope differential readings, using derived alignment and granularity parameters to dynamically regenerate cryptographic keys without storing them. Using five Raspberry Pi 5 devices, each with four MPU-9250 sensors, our experiments demonstrate effective device identification and assess the resilience of motion-regenerated ICMetric-based keys to brute-force attacks.

1 Introduction

The Internet of Things (IoT) connects a vast number of devices, interacting with both digital and physical environments. Ensuring secure and accurate identification of these devices is a critical challenge. Conventional security methods typically depend on stored cryptographic keys, which may be extracted or cloned. The ICMetric technology addresses this challenge by generating device identities based on inherent and unique device features.

The earlier work by Tahir et al. (2015) demonstrated that accelerometer-based signatures could be used for device identification in static conditions where sensors (Shimmer, 2013) were placed on a flat surface. The purpose of inertial sensors is to measure acceleration and angular velocity, for that reason, this kind of sensors are typically mounted in devices that are moving. Therefore, in contrast to Tahir et al. (2015), this study explores dynamic device identification, where sensors are subject to continuous motion and vibration, introducing noise, and causing the distributions of raw sensor readings to overlap significantly, making device-level differentiation more challenging. We propose an ICMetric key-generation scheme that derives alignment (offset) and granularity (range) parameters from inter-sensor differential readings which regenerates cryptographic keys on demand without persistent storage. Crucially, we analyse how probability-density-based range segmentation (percentile/PDF thresholds) affects key granularity, entropy, and reproducibility in motion, quantifying key-space growth, Mean Key Index Range (MKIR), and correct-key rates across thresholds.

The proposed scheme has conceptual similarities to a Physically Unclonable Function (PUF). However, our proposal does not require the inclusion of any additional hardware component. Rather, the use of inertial sensors and the training period during which a behavioural profile is obtained grants it a location-bounding property, causing regeneration of an invalid key when the device is physically intercepted and moved away from its original environment.

Contributions:

1. A device-identification study using inter-sensor differential features collected during real-world vehicle motion.
2. An ICMetric key-generation method for inertial sensor data that uses alignment and granularity parameters to regenerate keys without storing them.
3. A quantitative key-generation analysis comparing percentile-based granularity estimation, reporting

[a] https://orcid.org/0009-0002-2343-3362
[b] https://orcid.org/0000-0001-5590-0880

effects on key space, MKIR, and correct-key rate (see Table 2).

The paper is organised as follows. Section 2 reviews related work on ICMetric-based device identification and key generation, with a focus on inertial sensor approaches. Section 3 describes the experimental setup, including the hardware configuration, data collection procedure, and feature preparation. Section 4 describes the methodology used for device identification. Section 5 presents the proposed ICMetric key-generation method based on dynamic inertial sensor data, detailing feature extraction, key composition, and dynamic regeneration. Section 6 describes the methodology for analysis of the percentile-based range selection impact on key granularity, entropy, and reproducibility. Section 7 describes the key generation analysis procedure. Section 8 reports the experimental results, covering both device identification performance and ICMetric key-generation metrics. Section 9 discusses security considerations, attack surfaces, and limitations of the proposed approach. Finally, Section 10 concludes the paper and outlines directions for future work.

2 Related Work

Over the past two decades, ICMetric-based approaches have been applied across a range of domains, including generic embedded systems (Hopkins et al., 2007; Zhai et al., 2013a,b), IoT devices (Tahir and McDonald-Maier, 2012; Tahir et al., 2016), cloud environments (Tahir et al., 2013a; Ye et al., 2013; Baba et al., 2019), healthcare applications (Papoutsis et al., 2009; Kovalchuk et al., 2011, 2012b), and automotive systems (Alheeti and McDonald-Maier, 2016).

The idea governing ICMetrics is to represent a device's identity using its intrinsic characteristics, much as a biological organism's identity can be represented by biometrics (e.g. DNA, fingerprints, facial structure, movement patterns, retinal composition, ear structure). Similar to the wide variety of potential biometric features, there is also a broad range of potential ICMetric features.

Hopkins et al. (2007) identified the following potential features: addresses of program instructions (program counter), data transaction addresses, sizes and types, page faults, cache hits and misses. It is worth emphasising that monitoring data transactions broadens the monitoring scope to cover system interaction with external devices that use memory-mapped peripherals. Kovalchuk et al. (2012a) explored a combination of program counter, status register and stack pointer. Zhai et al. (2013a) used program counter values and their corresponding clock cycles per instruction (CPI) values which vary upon instruction types and allow to capture behavioural profile over long periods of time (by computing average CPI over a specific period of time). Aside from previously incorporated hardware and software characteristics, Tahir et al. (2013b) incorporated a random session ID assigned by a trusted party into the ICMetric key responsible for secure communication over a network. Alheeti and McDonald-Maier (2016) demonstrated an ICMetric system where external magnetometer readings were used to secure a network of self-driving vehicles. Similarly, Tahir et al. (2015) exploited accelerometer bias as a device fingerprint under stationary conditions. Hadrich et al. (2025) and Zhang et al. (2025) demonstrated PUF key generation based on inertial sensors data, however in each case sensors were stationary. To date, ICMetric and PUF research based on inertial sensor information has been confined to static scenarios, leaving the applicability of such techniques under dynamic motion unexplored. In dynamic environments, absolute sensor bias becomes ineffective, motivating the need for alternative features based on differential multi-sensor data. This work addresses this gap by extending the ICMetric inertial sensor approach to dynamically collected data and by enabling cryptographic key generation from real-time sensor characteristics.

3 Experimental Setup

3.1 Hardware Configuration

Five Raspberry Pi 5 devices were employed, each connected to four MPU-9250 accelerometers through a TCA9548A I^2C multiplexer (Fig. 1). The devices were powered and placed in the boot of a car near the rear axle. Hardware included:

- 5 × Raspberry Pi 5 devices
- 1 × TCA9548A I^2C multiplexer per device
- 4 × MPU-9250 sensors per device

Figure 1: Experimental setup.

- Laptop running a TCP server, initiating data collection

Contrary to the approach of Zhang et al. (2025), who used a 2D array of sensors operated by a single controller (driving a row decoder and column multiplexer), in this case, five separate Raspberry Pi devices operate their respective sensors in parallel. This was crucial considering the dynamic environment, as any delay between reading individual sensors could impact the collected data (due to changes in vehicle speed and direction).

3.2 Data Collection Procedure

The car was driven on local roads while continuously recording acceleration and gyroscope readings. Each device collected 100,000 samples. To mitigate positional bias, the devices were rotated every 20,000 samples (Fig. 2) so that each device occupied the same positions. This dataset was used for both studies presented in this paper: device identification and key generation analysis.

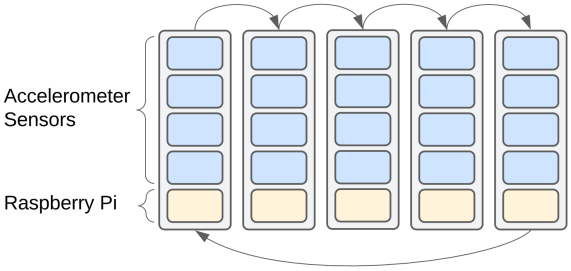

Figure 2: Positional rotation visualised.

3.3 Data Preparation

Each collected sample consists of raw acceleration and gyroscope readings from four sensors in 3 directions: x, y, and z. To mitigate the impact of movement, inter-sensor differences were computed:

$$\Delta_{ij}(t) = a_i(t) - a_j(t) \quad (1)$$

The resulting difference vectors consisted of 18 features, including differences between sensors: 0-1, 1-2, 2-3, across 6 features (acceleration in x, y, z directions, and gyroscope x, y, z readings), and which were used as an input for both identification and key generation.

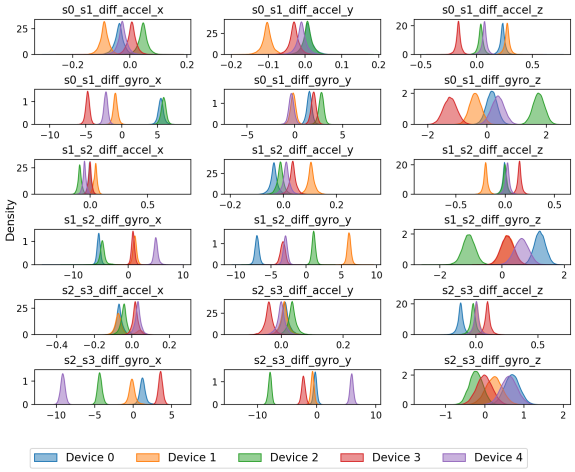

Figure 3: Distribution of collected features.

4 Device Identification from Dynamic Sensor Data

Figure 3 presents kernel density estimates (with bandwidth parameter set to 0.1) of collected features distributions. It can be seen that the differences between readings of individual sensors attached to separate devices are consistently different.

Figure 3 illustrates the distributions of pairwise sensor-difference features computed from sensors attached to each Raspberry Pi device. Each colour in the figure corresponds to a different Raspberry Pi, with each device connected to four sensors. From these sensors, a total of 18 features were derived to characterise each measurement. To evaluate whether a Raspberry Pi can be correctly identified from a given sensor measurement, we formulated a device recognition task in which each sample is represented by this 18-dimensional feature vector. Several classification algorithms were evaluated for this task, their performance is reported in Section 8. The evaluated algorithms include:

- Gaussian multivariate distribution (probability density function)
- K-Nearest Neighbour (KNN)
- Support Vector machine (SVM)
- Decision Tree (DT)
- Random Forest (RF)
- Neural Network (NN) of three layers

For each algorithm, the 500,000 total samples were divided randomly 80:20 into training (400,000) and testing (100,000) datasets. This was done 5 times using K-fold cross-validation method. Computed metrics consisted of accuracy, precision, recall,

and F1 score averaged across all 5 folds. Due to negligible differences among different metrics (equivalent to rounding error), only the accuracy is shown in the results section.

5 ICMetric Key Generation from Dynamic Sensor Data

The ICMetric technology relies on generating reproducible identifiers derived from dynamically varying inherent device features, rather than storing cryptographic keys permanently. In this work, inertial sensor readings collected during motion are used to generate an ICMetric-based key.

5.1 Feature Extraction

For each pair of MPU9250 sensors connected to the same Raspberry Pi, the difference signal $\Delta_{ij}(t)$ was computed across all axes of accelerometer and gyroscope readings. From this, we determine:

- **Granularity (range size):** derived from the central portion of the feature distribution, bounded by a chosen percentile interval to exclude outliers. Specifically, the lower and upper bounds are defined as the quantiles:

$$x_L = Q\left(0.5 - \frac{p}{2}\right), \quad x_U = Q\left(0.5 + \frac{p}{2}\right) \tag{2}$$

  where $Q(\cdot)$ denotes the empirical quantile function and p is the selected percentile (e.g., $p = 0.95$ for the 95% central range). The *granularity* is then given by the range between these quantile boundaries:

$$G_{ij} = x_U - x_L \tag{3}$$

  This value represents the effective spread of the central p proportion of $\Delta_{ij}(t)$, mitigating the detrimental effect of outliers on the key entropy, while preserving most of the sensor variability.

- **Alignment (offset):** Distance between 0 and the closest lower multiple of G_{ij} for x_L, as depicted in Figure 4. The purpose of alignment is to ensure that the granularity windows are aligned with the distribution centre of a given feature, ensuring that values collected from a specific sensor will likely fall into a specific window (producing the same key part). Compared to the selection of specific bits within the binary representation of the collected value used by Zhang et al. (2025), the use of alignment establishes more precise boundaries that are not restricted to being divisible by the powers of 2.

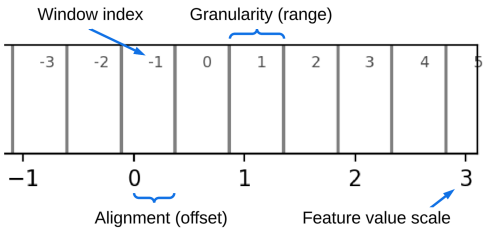

Figure 4: Granularity and alignment visualised.

5.2 Key Composition

The ICMetric key is generated from the concatenation of quantised feature windows (indices):

$$K = f\left(\text{Index}_{\text{window}}(\text{Alignment}, \text{Granularity})\right) \tag{4}$$

Each window index corresponds to a quantised range of $\Delta_{ij}(t)$ values defined by G_{ij}. The key space expands exponentially with the number of sensors and axes, making brute-force recovery infeasible when sufficient features are used.

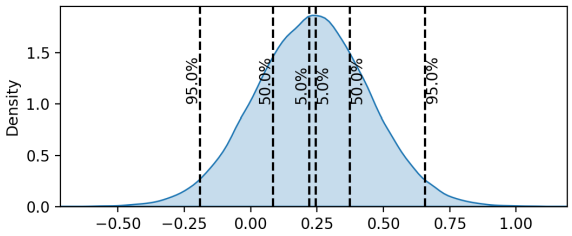

Figure 5: Percentile bounds visualisation.

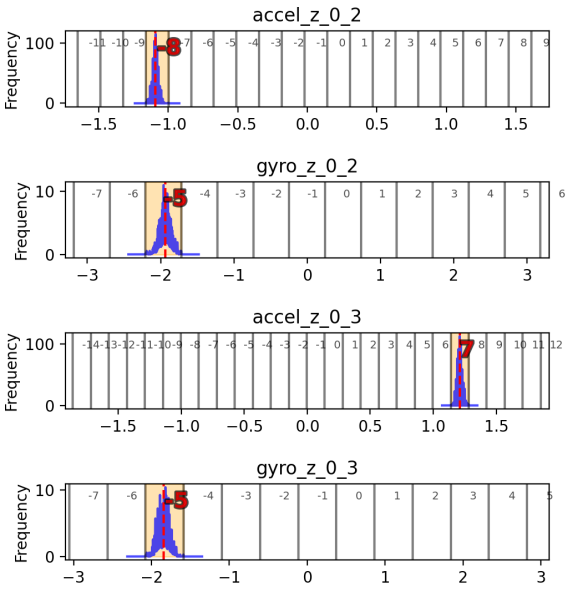

Figure 6: Key generation visualisation, showing how key parts (-8, -6, 7, -5) are derived from data of two features (using 100% percentile for quantisation).

5.3 Dynamic Regeneration

The key is not stored; it is regenerated dynamically using live accelerometer and gyroscope readings. Only the alignment and granularity parameters are stored locally, allowing the device to reproduce the same key under normal operational conditions but produce a different key when a sensor is substituted (which is evident from Figure 3).

6 Impact of Percentile-Based Ranges on Key Granularity

The proposed key generation framework determines feature ranges using percentile-based quantile boundaries rather than absolute limits. This approach focuses on the central portion of the feature distribution. The range defined by these percentiles serves as the basis for computing the *granularity* used in ICMetric key generation.

6.1 Motivation

In dynamic environments, such as vehicular motion, sensor readings can include high-amplitude transient values that distort the statistical spread of features. Relying on absolute extremes may therefore produce coarse range definitions, reducing entropy per feature. To address this, range segmentation is defined by a fixed percentile window (e.g., 90% or 95%) around the median of each feature distribution, improving discriminative resolution.

6.2 Expected Impact

By varying the percentile parameter, it is possible to control the trade-off between reproducibility and entropy:

- **Key entropy:** Smaller percentile windows result in finer quantisation, allowing more sensitive differentiation between devices and increasing resistance to brute-force key recovery.
- **Stability:** Larger percentile windows improve robustness to noise and temporal variation but reduce entropy.

When the percentile parameter is set to values lower than 1, the derived key is no longer guaranteed to be reproducible from a single observation. In this regime, sensor variability may cause individual feature values to fall outside the expected window, leading to inconsistent key indices. As a result, key generation may require multiple independent attempts, with each attempt producing a candidate key. A voting-like or majority-based mechanism can then be employed across repeated measurements to identify the most frequently occurring key as the correct one.

This analysis quantifies how percentile-based range segmentation influences key space size, mean key index range (MKIR), and correct-key rate across different probability thresholds (e.g., 95%, 90%, 80%), providing insight into the balance between entropy and stability in dynamic key generation.

7 Key Generation Analysis Procedure

All collected samples (each containing 18 features) were split into 90% training (450,000) and 10% testing (50,000) datasets. For each evaluated percentile, based on the training dataset, a granularity (range) and alignment (offset) value were computed. In conjunction with median feature values from the training dataset, window indices were produced and formed parts of the key, producing concatenated and hashed baseline keys. Equation 5 specifies how a part of a key would be produced for a single feature from a training dataset. In the case of key regeneration (using testing dataset), the key part would be produced from a single sample, meaning that the raw feature value would take the place of the median value in the Equation 5.

$$\text{key\_part} = \left\lfloor \frac{\text{median\_value} - \text{offset}}{\text{range}} \right\rfloor \tag{5}$$

For every sample consisting of eighteen features, a key part (window index) was generated and concatenated into a single key, represented as a string (for convenient representation/understanding), and hashed to provide uniformity.

8 Results

This section reports the results of two related but distinct experimental studies. First, we evaluate the ability to identify individual Raspberry Pi devices under dynamic motion using inter-sensor differential features. Second, using the same features, we assess the security and reproducibility properties of regenerated ICMetric keys, with particular emphasis on how percentile-based granularity selection impacts key space size, entropy, and correct-key regeneration rates.

8.1 Device identification

Table 1 summarises the device identification performance of all evaluated classifiers under dynamic sensing conditions. Despite the presence of significant motion-induced noise and overlapping raw sensor distributions, all methods achieve near-perfect classification accuracy, demonstrating that inter-sensor differential features remain highly discriminative even during real-world vehicle motion.

Table 1: Average Incorrect Device Predictions

	Gauss[a]	KNN[b]	SVM[c]	DT[d]	RF[e]	NN[f]
	4.0	0.2	0.0	2.60	0.0	0.8
(%)	0.004%	0.0002%	0.0%	0.0026%	0.0%	0.0008%

[a] Multivariate normal PDF classifier (SciPy).
[b] k-Nearest Neighbours with $k = 3$ (scikit-learn).
[c] Support Vector Machine with default parameters (scikit-learn).
[d] Decision Tree with default parameters (scikit-learn).
[e] Random Forest with 100 estimators (scikit-learn).
[f] TensorFlow/Keras NN with two hidden layers (64 neurons each, ReLU) and softmax output.

While the overall misclassification rates remain extremely low across all models, some differences between classifiers are still evident. Within the parameters of the test undertaken, both the Support Vector Machine and Random Forest achieve zero incorrect predictions on the evaluated dataset, demonstrating perfect classification performance. In contrast, the Gaussian model, KNN, Decision Tree, and Neural Network exhibit small but non-zero error rates, with the Gaussian and Decision Tree models showing comparatively higher misclassification counts.

Although these error rates are negligible from a conventional machine-learning perspective, it is important to note that even rare misclassifications are unacceptable in the context of ICMetric-based security systems. A single incorrect classification may result in key regeneration failure, leading to authentication denial or system lockout, which is unacceptable in practice. This observation motivates the preference for classifiers with strictly zero observed error when inertial features are subsequently reused for cryptographic key derivation.

8.2 ICMetric key-generation

Table 2 presents a quantitative analysis of ICMetric key-generation characteristics as a function of the selected percentile range used to define feature granularity. By progressively narrowing the percentile window, the effective feature range decreases, resulting in finer quantisation, increased Mean Key Index Range (MKIR), and exponential growth of the theoretical key space.

Table 2: Summary of Key Generation Metrics Across Percentiles

Perc.	Avg. Range	MKIR[a]	Possible Keys	KBS[b]	Correct keys[c]
1.0	5.2452	3.56	4.98×10^{9}	33	99.98%
0.99	0.9935	5.89	1.28×10^{13}	44	89.0%
0.95	0.6537	7.33	3.45×10^{14}	49	56.94%
0.90	0.5171	8.78	6.32×10^{15}	53	33.43%
0.80	0.3816	10.67	2.31×10^{17}	58	11.41%
0.70	0.3011	13.22	1.09×10^{19}	64	3.5%
0.60	0.2407	15.89	1.96×10^{20}	68	0.76%
0.50	0.1906	20.00	1.20×10^{22}	74	0.1%
0.40	0.1465	25.00	4.96×10^{23}	79	0.0%

[a] Mean Key Index Range.
[b] Key bit-size
[c] The percentage of keys in the testing dataset matching the key generated from the training dataset.
The correct key had the highest frequency among other keys.
The correct key was present in the tested samples, but there was a more frequent incorrect key.
The correct key was not present in the tested samples.

The results reveal a clear, though not surprising, trade-off between key entropy and reproducibility. Higher percentile values (e.g., 95% and 90%) provide greater stability, yielding higher correct-key regeneration rates, but at the cost of reduced entropy and smaller effective key spaces. Conversely, lower percentile thresholds significantly increase key bit-length and brute-force complexity; however, this comes at the expense of reproducibility, with correct-key rates rapidly approaching zero below the 80% threshold. These findings demonstrate that percentile-based granularity selection is a critical design parameter: aggressive entropy maximisation alone is insufficient if it compromises reliable key regeneration under dynamic conditions.

9 Discussion

9.1 Security Considerations and Attack Surface Analysis

While the ICMetric approach eliminates the need for stored cryptographic keys, reducing the risk of digital key extraction, it is still essential to evaluate potential attack vectors and the overall resilience of the system.

9.1.1 Brute-Force Attack Feasibility

NIST guidance (SP 800-131A Rev.2 / SP 800-57; Barker and Roginsky 2019) identifies a practical baseline security strength of 112 bits, with 128 bits rec-

ommended for new or long-lived systems. The effective key-widths reported in Table 2 are, for many percentile settings, well below the 112-bit baseline and thus insufficient by NIST standards.

To meet NIST recommendations, the effective entropy of the regenerated ICMetric key must be increased (or conservatively estimated via min-entropy after helper-data disclosure). Practical ways to achieve this include: (1) adding more sensors or axes to increase independent features (considering the average entropy of tested sensors, 10 MPU9250 sensors would be sufficient to generate a key surpassing the 112-bit key threshold suggested by NIST using 0.95 percentile), (2) including additional sensor modalities such as magnetometer, and (3) incorporating device-intrinsic characteristics (e.g. on-chip noise/temperature fingerprints, OS-related features).

9.1.2 Replay Attacks

A fundamental security property of our scheme is that features are formed from inter-sensor differences and their quantised indices, not from raw motion traces. The quantised indices and the alignment/granularity parameters that determine them are not observable to an external adversary who only records the device's motion. Consequently, passive recording of a device's movement (for example, logging the accelerations experienced during a drive) does not provide the attacker with the information necessary to reproduce the ICMetric key: identical external motion applied to another device will not yield the same differential indices.

9.1.3 Physical and Side-Channel Attacks

Although the ICMetric key itself is not stored, the system remains susceptible to physical and hardware-level attacks. If an attacker gains physical access to the device, exposed sensor communication lines (e.g., I^2C connections between the microcontroller and inertial sensors) could be probed to directly read sensor values, enabling the reproduction of ICMetric features externally. To reduce exposure to this threat, sensors should ideally be integrated within the same package as the processing unit, ensuring that raw sensor data cannot be intercepted easily.

This countermeasure reduces the risk of compromise; however, recent advances in hardware probing attacks suggest that additional protections, such as enclosing the SoC within a secure enclave, or a more elaborate system, such as "Reconfigurable Field Effect Transistor" introduced by Parvin et al. (2025), may be necessary. Several studies have demonstrated the ability to infer or directly observe the internal state of signals and registers in ASIC and FPGA devices, particularly in architectures fabricated at 28 nm process nodes, such as those commonly used in Kintex-7 Hori et al. (2012) and Artix-7 Moradi and Schneider (2016) devices, using power analysis, electromagnetic radiation and more recently introduced optical probing Tajik et al. (2017).

9.2 Summary

Overall, while brute-force recovery of the key is computationally impractical, the physical security of the system remains critical. Embedding the sensors directly within the microcontroller die or within a tamper-resistant secure module would significantly strengthen the protection of sensor-derived secrets. Future ICMetric implementations should therefore consider integrated sensor–processor architectures to eliminate physical exposure and enhance resistance to side-channel observation.

10 Conclusion

This paper demonstrated that inertial sensor–based device identification remains effective under dynamic conditions by exploiting inter-sensor differential features. Despite significant motion-induced noise, near-perfect identification accuracy was achieved, confirming that relative sensor relationships provide stable and discriminative device fingerprints during real-world operation.

Using the same features, we proposed an ICMetric key-generation approach based on percentile-derived alignment and granularity parameters, enabling keys to be regenerated without permanent storage. Our results show a clear trade-off between key entropy and reproducibility: narrower percentile ranges substantially increase key space and brute-force complexity but reduce correct-key regeneration rates. While the derived keys are device-dependent and resistant to replay, the effective key bit-width achieved with the evaluated feature set remains below current NIST recommendations. Achieving cryptographic key strengths compliant with current security recommendations requires increasing the number of sensors and/or independent features used in the ICMetric key generation.

ACKNOWLEDGEMENTS

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/Y030168/1.

REFERENCES

Alheeti, K. M. A. and McDonald-Maier, K. (2016). An intelligent intrusion detection scheme for self-driving vehicles based on magnetometer sensors. In *2016 International Conference for Students on Applied Engineering (ISCAE)*, pages 75–78. IEEE.

Baba, S. D., Yadav, S., and Howells, G. (2019). Sortalgometrics: Identification of cloud-based server via a simple algorithmic analysis. In *2019 Eighth International Conference on Emerging Security Technologies (EST)*, pages 1–6. IEEE.

Barker, E. and Roginsky, A. (2019). NIST Special Publication 800-131A Revision 2: Transitioning the Use of Cryptographic Algorithms and Key Lengths. Technical Report NIST SP 800-131A Rev. 2, National Institute of Standards and Technology.

Hadrich, W., Zimmermann, L., Tritschler, P., and Sikora, A. (2025). A novel puf key generation method via variable-length subkeys: An application with inertial mems sensors. In *Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, pages 27–28. ACM.

Hopkins, A., McDonald-Maier, K., Papoutsis, E., and Howells, W. (2007). Ensuring data integrity via icmetrics based security infrastructure. In *Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007)*, pages 75–81. IEEE.

Hori, Y., Katashita, T., Sasaki, A., and Satoh, A. (2012). Electromagnetic side-channel attack against 28-nm fpga device.

Kovalchuk, Y., Howells, W., Hu, H., Gu, D., and McDonald-Maier, K. (2012a). A practical proposal for ensuring the provenance of hardware devices and their safe operation. In *7th IET International Conference on System Safety, incorporating the Cyber Security Conference 2012*, pages 1–6.

Kovalchuk, Y., Hu, H., Gu, D., McDonald-Maier, K., Newman, D., Kelly, S., and Howells, G. (2012b). Investigation of properties of icmetrics features. In *2012 Third International Conference on Emerging Security Technologies*, pages 115–120. IEEE.

Kovalchuk, Y., McDonald-Maier, K., and Howells, G. (2011). Overview of icmetrics technology-security infrastructure for autonomous and intelligent healthcare system. Technical report.

Moradi, A. and Schneider, T. (2016). Improved side-channel analysis attacks on xilinx bitstream encryption of 5, 6, and 7 series. volume 9689, pages 71–87.

Papoutsis, E., Howells, G., Hopkins, A., and McDonald-Maier, K. (2009). Ensuring secure healthcare communications via icmetric based encryption on unseen devices. In *2009 Symposium on Bio-inspired Learning and Intelligent Systems for Security*, pages 113–117. IEEE.

Parvin, S., Jha, C. K., Torres, F. S., and Drechsler, R. (2025). True-polytronik: Securing circuits against laser logic state imaging attack using rfet. In *2025 38th International Conference on VLSI Design and 2024 23rd International Conference on Embedded Systems (VLSID)*, pages 37–42. IEEE.

Shimmer (2013). *Shimmer User Manual Revision 2RK*. Shimmer, Dublin.

Tahir, H., Tahir, R., and McDonald-Maier, K. (2013a). A novel private cloud document archival system architecture based on icmetrics. In *2013 Fourth International Conference on Emerging Security Technologies*, pages 102–106. IEEE.

Tahir, H., Tahir, R., and McDonald-Maier, K. (2015). Securing mems based sensor nodes in the internet of things. In *2015 Sixth International Conference on Emerging Security Technologies (EST)*, pages 44–49. IEEE.

Tahir, R., Hu, H., Gu, D., McDonald-Maier, K., and Howells, G. (2013b). A scheme for the generation of strong icmetrics based session key pairs for secure embedded system applications. In *2013 27th International Conference on Advanced Information Networking and Applications Workshops*, pages 689–696. IEEE.

Tahir, R. and McDonald-Maier, K. (2012). An icmetrics based lightweight security architecture using lattice signcryption. In *2012 Third International Conference on Emerging Security Technologies*, pages 135–140. IEEE.

Tahir, R., Tahir, H., McDonald-Maier, K., and Fernando, A. (2016). A novel icmetric based framework for securing the internet of things. In *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pages 469–470. IEEE.

Tajik, S., Lohrke, H., Seifert, J.-P., and Boit, C. (2017). On the power of optical contactless probing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1661–1674. ACM.

Ye, B., Howells, G., and Haciosman, M. (2013). Investigation of properties of icmetric in cloud. In *2013 Fourth International Conference on Emerging Security Technologies*, pages 107–108. IEEE.

Zhai, X., Appiah, K., Ehsan, S., Cheung, W. M., Hu, H., Gu, D., McDonald-Maier, K., and Howells, G. (2013a). A self-organising map based algorithm for analysis of icmetrics features. In *2013 Fourth International Conference on Emerging Security Technologies*, pages 93–97. IEEE.

Zhai, X., Appiah, K., Ehsan, S., Hu, H., Gu, D., McDonald-Maier, K., Cheung, W. M., and Howells, G. (2013b). Application of icmetrics for embedded system security. In *2013 Fourth International Conference on Emerging Security Technologies*, pages 89–92. IEEE.

Zhang, W., Wang, S., Wang, P., Fu, Q., and Li, X. (2025). Design of accelerometers-based puf for internet of things security. *IEEE Sensors Journal*, 25:32984–32992.