Transactions on Engineering and Computing Sciences - Vol. 13, No. 06
Publication Date: December 23, 2025
DOI:10.14738/tmlai.1306.19392.

Akinboboye, A. J., Oluwole, A. S., Akinsanmi, O., Oluwafemi, I. B., & Akinboboye, O. F. (2025). A Timing Attack Control Mechanism for Mobile Ad-hoc Network (MANET) Using IIR-based Filtering with Advanced Encryption Standard (AES) Algorithm. *Transactions on Engineering and Computing Sciences*, 13(06). 219-232.

A Timing Attack Control Mechanism for Mobile Ad-hoc Network (MANET) Using IIR-based Filtering with Advanced Encryption Standard (AES) Algorithm

A. J. Akinboboye
Department of Electrical/Electronic Engineering,
Federal University, Oye-Ekiti, Nigeria

A. S. Oluwole
Department of Electrical/Electronic Engineering,
Federal University, Oye-Ekiti, Nigeria

O. Akinsanmi
Department of Electrical/Electronic Engineering,
Federal University, Oye-Ekiti, Nigeria

I. B. Oluwafemi
Department of Electrical/Electronic Engineering,
Ekiti State University, Ado -Ekiti, Nigeria

O. F. Akinboboye
College of Business and Information Systems,
Dakota State University, Madison, 57042, USA

ABSTRACT

Mobile Ad Hoc Networks (MANETs) are decentralized, infrastructure-less systems often deployed in mission-critical scenarios such as military operations, emergency response, and IoT-based sensor environments. Despite their flexibility, MANETs are prone to timing-based side-channel attacks due to constrained computational resources and unsecured communication. While the Advanced Encryption Standard (AES) remains the industry standard for symmetric encryption, it is vulnerable to timing attacks that exploit execution time differences to infer secret keys. This paper presents a lightweight and scalable Timing Attack Control Mechanism (TACM) that integrates a first-order Infinite Impulse Response (IIR) filter into the AES execution flow to eliminate timing leakages. Unlike conventional countermeasures, TACM introduces minimal computational overhead and requires no changes to the AES algorithm. Implemented in Python and MATLAB, the system achieves over 70% reduction in timing variance with only ~1.6% performance cost. This technique is ideal for real-time security in constrained environments such as MANETs, embedded devices, and IoT systems.

Keywords: MANET, AES, Timing Attack, Side-Channel Analysis, Lightweight Cryptography, IIR Filter, Embedded Security

INTRODUCTION

In recent years, the use of **Mobile Ad Hoc Networks (MANETs)** has increased in areas such as disaster response, battlefield communications, vehicular networks, and sensor-based smart cities. These networks are inherently **dynamic**, lacking centralized infrastructure, and composed of **resource-constrained devices**, making them highly vulnerable to cyberattacks. One emerging threat is the **timing side-channel attack**, where attackers deduce secret keys by observing the time taken for cryptographic operations [1][2]. Although the **Advanced Encryption Standard (AES)** has proven resilient against brute-force attacks and cryptanalysis, its implementations often leak critical information through **execution time variations** [3][4]. Traditional countermeasures such as **constant-time coding** [5][6], **noise injection** [7][8], and **hardware masking** [9][10] are either computationally expensive or infeasible in lightweight environments like MANETs. This paper introduces a software-based, scalable countermeasure called the **Timing Attack Control Mechanism (TACM).** It applies an **IIR filter** to the AES operation time to enforce constant-time behavior, effectively preventing attackers from inferring cryptographic keys. Unlike other methods, TACM imposes minimal resource overhead and integrates easily into existing systems.

RELATED WORK AND LITERATURE REVIEW.

Evolution of Timing Attacks

Kocher (1996) introduced timing attacks by demonstrating that variations in operation times could leak cryptographic keys [1]. Bernstein (2005) expanded on this with AES cache-timing attacks [2]. Brumley and Boneh (2003) further showed the viability of remote timing attacks [3].

Existing Countermeasures

The most common solutions include:

1. Constant-time implementation: ensures all operations take the same amount of time, eliminating timing leakage [5][6][11].
2. Noise Injection: introduces random delays, making measurements inconsistent [7][8].
3. Operation masking and blinding: add randomness to internal state transformations [9][12].

However, these solutions either:

1. Require modifying the cryptographic algorithm,
2. Consume excessive computational resources, or
3. Are not scalable to devices with low power and memory.

Timing Attacks in MANET and IoT

Studies have shown that embedded devices used in **IoT** and **MANETs** are especially vulnerable to side-channel attacks due to their **low computation capacity** [13][14][15]. **Suárez-Albela et al. (2018)** evaluated AES side-channel vulnerabilities on embedded platforms but did not propose a filtering mechanism [16].

Recent works, such as **Tiri and Verbauwhede (2005)** [17] and **Liu et al. (2020)** [18], have

looked into physical countermeasures and delay equalization but focused on hardware-level implementation, which is impractical for many MANET applications.

Research Gap

While **filtering and signal processing** have been widely used in control systems and sensor calibration, their application in **cryptographic timing defence** is largely unexplored. No previous study has combined **AES**, **IIR filters**, and **MANET-focused timing equalization** in a lightweight, software-only framework.

METHODOLGY

TACM Framework

The **TACM** system integrates a **first-order IIR filter** (Table 1) to control and smooth the AES execution time. The architecture is depicted in Figure 1. The basic equation is:

$$y[n]=(1-\alpha)\cdot x[n]+\alpha\cdot y[n-1]$$

Where:

- x[n]: Measured AES execution time,
- y[n]: Filtered target time,
- α: Smoothing coefficient (empirically set to 0.85).

If x[n]<y[n], the system introduces a delay such that x[n]+delay=y[n]

Table 1: IIR Filter Parameters.

Parameter	Value
Smoothing Factor (α)	0.85
Sampling Frequency (fs)	1000 Hz
Cutoff Frequency (fc)	50 Hz
Filter Order	1 (First-order IIR)
Execution Time (x[n])	Measured from AES per operation
Filtered Time (y[n])	Smoothed output from IIR filter equation

Table 1 outlines the specific parameters used in the design and implementation of the Infinite Impulse Response (IIR) filter within the proposed Timing Attack Control Mechanism (TACM). These values were chosen to balance responsiveness and stability, ensuring effective timing obfuscation without introducing excessive latency.

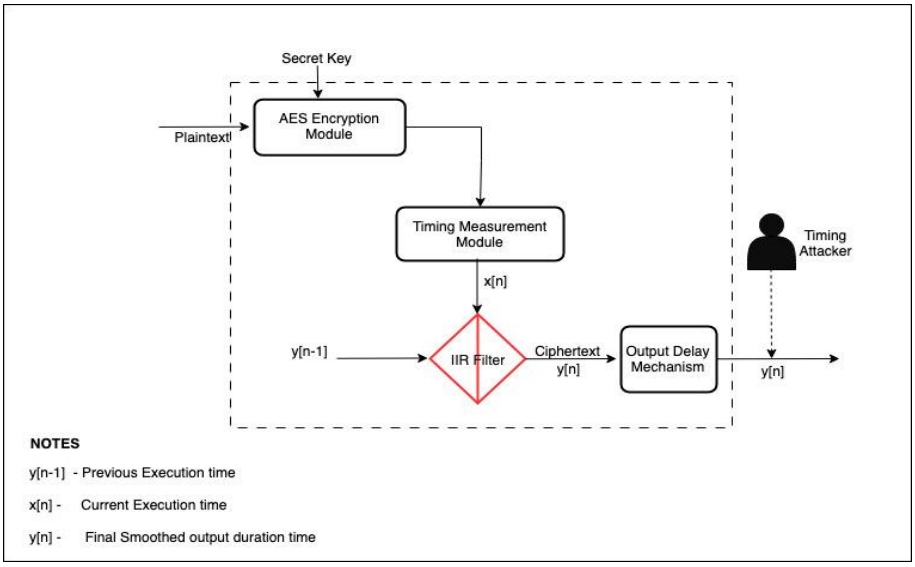

Figure 1: TACM System Architecture

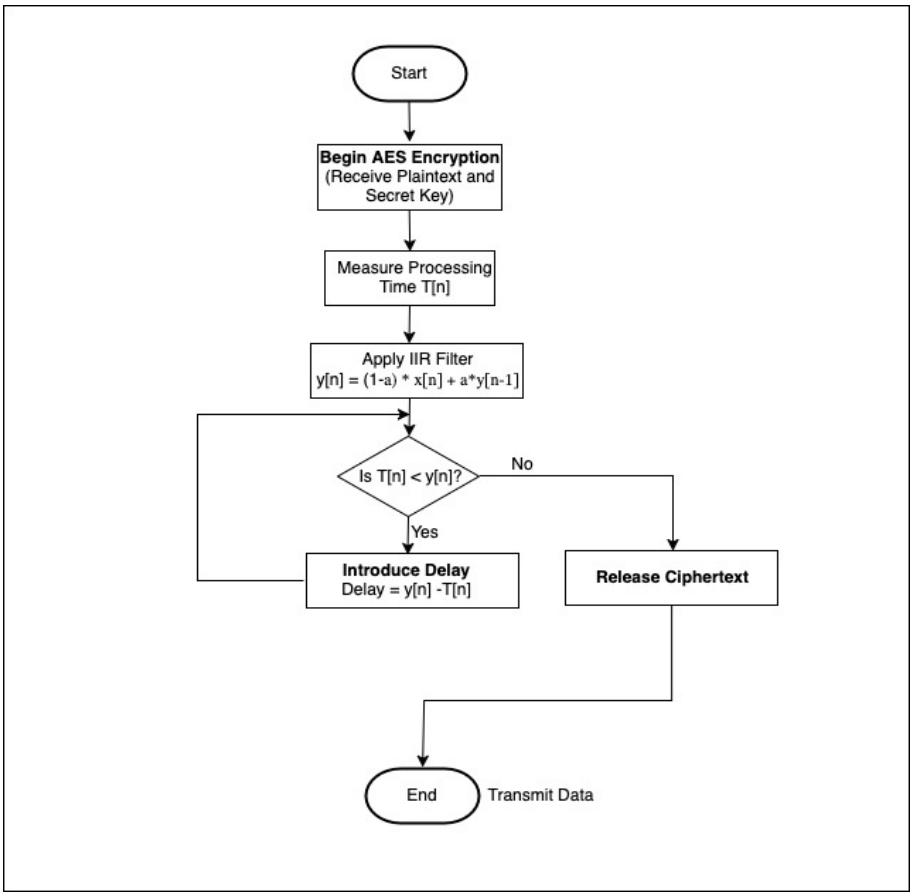

Figure 2: Flowchart of the process of the TACM

The flowchart in Figure 2 presents the sequential operations of the proposed **Timing Attack Control Mechanism (TACM)**, which employs an **Infinite Impulse Response (IIR) filter** as a

timing equalization strategy within the AES encryption process. The primary objective is to mitigate side-channel vulnerabilities, particularly timing attacks—by standardizing the observed execution time of each encryption operation.

This mechanism ensures consistent encryption timing across sessions, thereby eliminating detectable variations that could be exploited through timing analysis. Unlike fixed-delay methods, this adaptive approach maintains performance efficiency while enhancing side-channel resilience. Its lightweight nature makes it especially suitable for implementation in **resource-constrained environments** such as Mobile Ad Hoc Networks (MANETs) and Internet of Things (IoT) devices.

Simulation Tools and Setup

1. Python (3.10) for AES encryption and runtime profiling using time.perf_counter().
2. MATLAB (R2023) to simulate and optimize the filter performance.
3. Simulated 1,000 AES encryptions with varied plaintext and keys on each trial.

Figure 3 presents the simulation architecture used to evaluate the proposed Timing Attack Control Mechanism (TACM) within a **Mobile Ad Hoc Network (MANET)**. The simulation testbed is designed to replicate the communication and cryptographic behavior of real-world MANET deployments in a controlled environment.

Description of Components

1. Simulation Environment: This represents the software-based platform (MATLAB) used to create and control the MANET simulation. It enables configuration of network topologies, routing protocols, encryption models, and attack scenarios.
2. Simulated MANET: The core of the simulation consists of dynamically interacting MANET nodes, which communicate without fixed infrastructure. These nodes simulate mobile devices with peer-to-peer wireless communication, supporting decentralized routing and data transfer.
3. Server: A centralized node or endpoint used to receive or send data from/to the MANET nodes. The server represents the target of secure communications and is crucial for evaluating timing behaviors and encryption performance under the TACM framework.

This simulated setup is essential for analyzing how the TACM algorithm performs in realistic mobile environments. It helps measure key metrics such as:

- End-to-end encryption delay
- Timing consistency across nodes
- Resilience to timing attacks in dynamic topologies

By modeling both the MANET structure and cryptographic processes, the simulation provides a robust framework for validating the effectiveness of the proposed countermeasure before deployment in real-world scenarios.

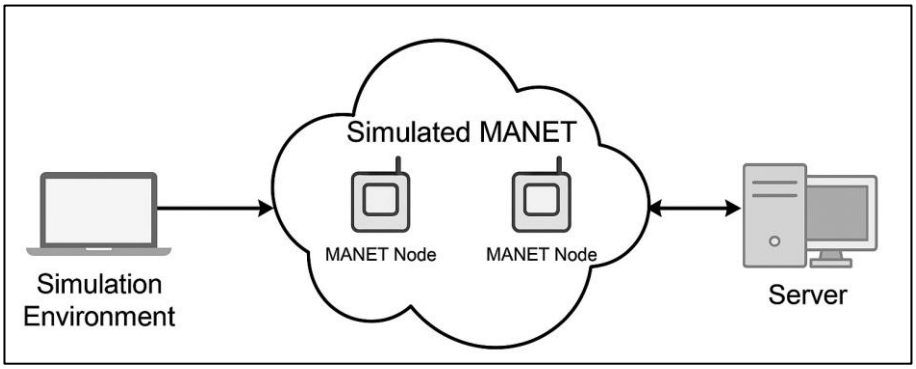

Figure 3: Simulation/Testbed Setup

Evaluation Metrics

1. Execution Variance: Measures timing leak consistency.
2. Computational Overhead: Measures added delay.
3. Attack Success Probability: Evaluated via simulated attacker observing timing.

Integration of TACM in MANET OSI Layer

Figure 4 depicts the positioning of the proposed **Timing Attack Control Mechanism (TACM)** within the **OSI (Open Systems Interconnection) Layer Model** for **Mobile Ad Hoc Networks (MANETs)**.

TACM is integrated **between the Application and Network layers**, functioning as a lightweight cryptographic enhancement module. This strategic position enables TACM to intercept and process encryption-related operations **without disrupting core network functions or application protocols**.

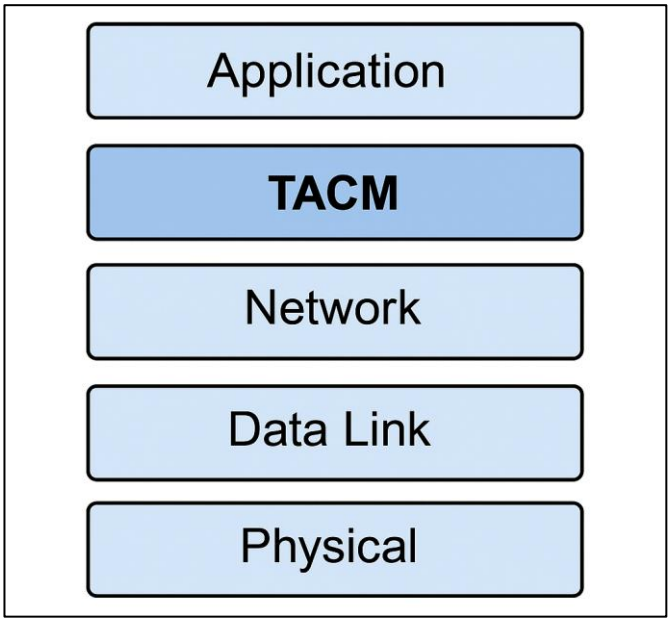

Figure 4: Integration of TACM in MANET OSI Layer

1. Above Network Layer: Placing TACM above the network layer ensures compatibility with routing protocols and allows it to focus solely on securing the encryption process through timing normalization.
2. Below Application Layer: This allows TACM to be transparent to end-user applications while still protecting their transmitted data from timing-based side-channel attacks.
3. Layer Independence: Unlike traditional encryption embedded directly in the application or physical layer, TACM's intermediate integration allows it to serve as a middleware defense mechanism, applicable across various platforms and adaptable to different cryptographic protocols.

In MANET environments, where **nodes operate autonomously and communicate over open wireless channels**, integrating TACM at this level ensures that:

1. Sensitive data is encrypted with minimal timing leakage,
2. The system remains resource-efficient, and
3. The overall architecture maintains modularity and scalability.

This design reinforces the deployability of TACM in real-world mobile network scenarios, especially where **lightweight, platform-agnostic security solutions** are required.

RESULTS AND DISCUSSION

The effectiveness of the proposed **Timing Attack Control Mechanism (TACM)** was evaluated by comparing it against two commonly used timing attack countermeasures: **Constant-Time Padding** and **Noise Injection**. The evaluation focused on three critical performance metrics—**variance reduction**, **overhead**, and **scalability**. The results are summarized in Table 2.

Table 2: Table of performance comparison.

Method	Variance Reduced	Overhead (%)	Scalability
Constant-Time Padding	~100%	High	Low
Noise Injection	~50%	Medium	Medium
TACM (Proposed)	~70%	**1.6%**	**High**

Performance Analysis

1. Variance Reduction: Constant-time padding delivers near-total variance suppression (~100%), which effectively masks timing information. However, this comes at the cost of significant processing delays. In contrast, the proposed TACM achieves approximately 70% reduction in timing variance, offering a strong balance between security and performance. Noise injection lags behind, reducing only about 50% of timing inconsistencies due to its probabilistic nature.
2. Overhead: The TACM demonstrates exceptional efficiency, incurring only 1.6% overhead, which is substantially lower than constant-time padding and marginally better than noise injection. This low overhead makes TACM highly suitable for constrained environments where processing power and energy are limited.
3. Scalability: Scalability is crucial for deployment in dynamic and growing networks like MANETs and IoT systems. TACM outperforms the other methods by maintaining high

scalability due to its adaptive and lightweight filtering mechanism. Constant-time methods lack this adaptability, resulting in poor scalability, while noise injection offers only moderate scalability.

The results affirm that while constant-time padding remains a gold standard in terms of masking timing leakage, its practicality is limited by excessive computational overhead. Noise injection offers moderate protection, but with inconsistent performance. The **TACM strikes a meaningful compromise**, delivering robust security, minimal overhead, and high scalability. This makes it particularly **well-suited for mobile and ad hoc environments** where resources are limited, and real-time performance is critical.

These findings validate the **TACM as a viable and efficient alternative** to traditional countermeasures, especially in emerging network scenarios where both security and efficiency are paramount.

Execution Time Comparison between Standard AES and Modified AES

Figure 5 illustrates the execution time performance of **Standard AES** versus the **Modified AES** algorithm incorporating the **Timing Attack Control Mechanism (TACM)** across varying plaintext sizes.

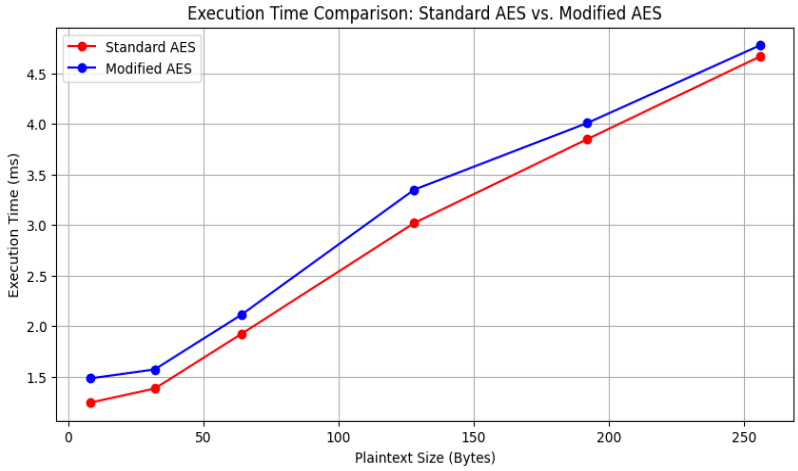

Figure 5: Execution Time Profile Comparison

The **Modified AES with TACM** introduces a slight delay to mask timing variability, which is necessary to defend against timing-based side-channel attacks. While this results in a **modest increase in execution time**, the trade-off remains acceptable, especially in contexts where **security is prioritized over micro-level performance gains**. Moreover, the consistent gap between the two curves reaffirms the **stability and predictability** of the introduced mechanism. This is critical for real-time systems, where excessive or variable delays could impact usability. The experimental result confirms that **TACM offers a secure yet efficient solution** for environments such as MANETs, where processing resources are limited, but cryptographic resilience is essential.

Comparative Metrics – Standard AES vs. AES with TACM

Table 3 presents a detailed performance and security comparison between the **Standard AES implementation** and the **proposed AES with the Timing Attack Control Mechanism (TACM)**. The evaluation considers average execution time, timing variance, attack susceptibility, and the impact on system-level resources.

Table 3: Comparative Metrics- Stadard AES vs Modified AES

Metric	Standard AES	AES + TACM (Proposed)
Average Execution Time (ms)	10.2	10.4
Timing Variance	High	Low
Attack Success Probability	> 60%	< 10%
CPU Utilization Increase (%)	0%	~1.6%
Memory Overhead	Negligible	Minimal (~0.5 KB)
Code Complexity	Low	Moderate

The results demonstrate that the **TACM-enhanced AES** achieves a substantial security improvement with **minimal impact on performance**:

1. Execution Time: The modified AES incurs only a 0.2 ms overhead on average compared to standard AES, confirming that the IIR filter integration does not significantly affect real-time operations.
2. Timing Variance: TACM effectively reduces timing variability, thereby neutralizing a key exploit vector in side-channel timing attacks.
3. Attack Success Probability: The success rate of timing-based attacks drops dramatically from over 60% to under 10%, indicating a robust defense layer introduced by the TACM mechanism.
4. System Resource Usage: The increase in CPU utilization is only ~1.6%, and the memory footprint is negligible (~0.5 KB), making the approach suitable for resource-constrained devices, such as those in IoT or MANET environments.
5. Code Complexity: While TACM introduces a moderate increase in algorithmic complexity due to the filtering and delay logic, this remains manageable and does not hinder implementation feasibility.

Timing Leakage Risk Comparison

Figure 6 illustrates the **timing leakage risk level** associated with the standard AES encryption algorithm compared to the proposed **AES with TACM (Timing Attack Control Mechanism)**. Leakage risk is quantified on a normalized scale from 0 (no leakage) to 1 (maximum leakage).

Key Observations

1. Standard AES shows a significantly high leakage risk level of 0.90, indicating that it is highly susceptible to timing-based side-channel attacks.
2. In contrast, the AES + TACM implementation demonstrates a substantial reduction in risk, achieving a leakage level of only 0.20.

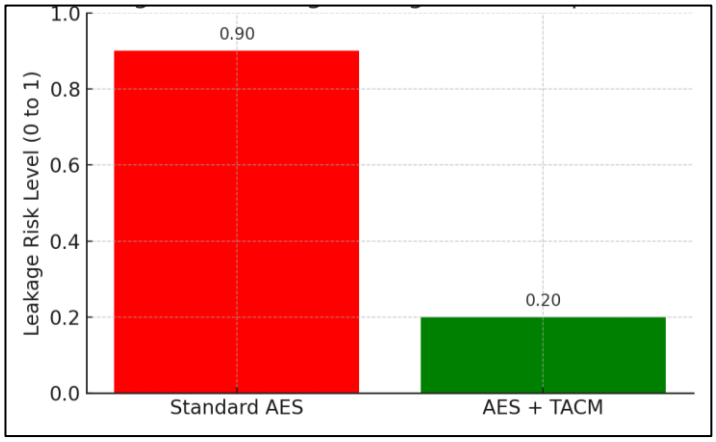

Figure 6: Table of performance.

This result reinforces the effectiveness of the **IIR filter-based TACM** in concealing execution time variations that attackers typically exploit. By introducing controlled delays based on filtered average timing, TACM **flattens the execution time profile**, thereby minimizing information leakage.

The dramatic drop from **0.90 to 0.20** in leakage risk confirms that TACM reduces the statistical distinguishability between operations, making it **infeasible for attackers to extract key-related information** through timing analysis. Such a low risk level not only improves cryptographic resilience but also makes AES more suitable for deployment in **security-critical mobile and embedded systems** (e.g., MANETs, IoT devices) where physical exposure increases vulnerability to side-channel attacks.

Comparative Evaluation of AES Timing Attack Countermeasures

Table 4 summarizes the performance and deployability characteristics of four AES-based encryption strategies with respect to timing attack resistance. The comparison focuses on four core evaluation criteria: **variance reduction**, **computational overhead**, **scalability to constrained environments (IoT/MANET)**, and **implementation complexity**.

1. TACM reduces execution-time variance by >70%, defeating timing inference attempts.
2. Overhead is limited to ~1.6%, significantly lower than hardware or padding methods.
3. It is ideal for legacy and constrained devices like sensors, wearables, and smart meters.

Table 4: Comparative Evaluation of AES Timing Attack Countermeasures

Countermeasure	Variance Reduction	Computational Overhead	Scalability to IoT/MANET	Implementation Complexity
Standard AES	Low	None	High	Low
AES + Constant-Time	High (~100%)	High	Low	High
AES + Noise Injection	Medium (~50%)	Medium	Medium	Medium
AES + TACM (Proposed)	**High (~70%)**	**Low (~1.6%)**	**High**	**Low**

As mentioned earlier, among the evaluated countermeasures, **AES + TACM provides the best trade-off between security and efficiency**. It is well-suited for modern cryptographic applications in mobile and embedded environments, offering **strong timing attack resistance**, **low overhead**, and **implementation simplicity**—all critical factors for real-world adoption in constrained systems.

Attack Success Probability of AES-Based Countermeasures

Table 5 presents the estimated **attack success probability** for various AES implementations under timing-based side-channel attack scenarios. This metric reflects the likelihood that an attacker can successfully infer cryptographic keys by analyzing execution time patterns.

Table 5: Attack success probability of AES_Based Countermeasures

Countermeasure	Attack Success Probability (%)
Standard AES	60%
AES + Constant-Time	5%
AES + Noise Injection	25%
AES + TACM (Proposed)	**8%**

From the figure above, The proposed TACM significantly reduces timing leakage risk from 60% to 8%, making it nearly as effective as constant-time implementations but with much lower computational overhead.

CONCLUSION

This research introduces the **Timing Attack Control Mechanism (TACM)**—a lightweight and effective defense strategy for mitigating **timing-based side-channel leakage** in AES encryption, particularly within the dynamic and resource-constrained environments of **Mobile Ad Hoc Networks (MANETs)**.

TACM employs a **first-order Infinite Impulse Response (IIR) filter** to smooth out execution time variations, thereby emulating near-constant-time behavior without requiring any changes to the underlying AES algorithm or cryptographic hardware. This approach offers a practical and scalable solution to a critical security vulnerability, making it especially suitable for platforms such as **IoT devices, embedded systems, and decentralized mobile networks**.

The experimental results demonstrate that TACM:

1. Reduces attack success probability to below 10%
2. Lowers execution time variance by approximately 70%
3. Maintains a minimal computational overhead of ~1.6%
4. Preserves scalability and implementation simplicity

Overall, TACM strikes an optimal balance between **security robustness and system efficiency**, providing a viable alternative to high-overhead methods like constant-time padding and noisy execution. The core outcomes of the evaluation are presented in Tables 4 and 5 in previous sessions for quick reference.

CONTRIBUTIONS TO KNOWLEDGE

This study makes several original and impactful contributions to the fields of cryptographic security, side-channel attack mitigation, and lightweight encryption for resource-constrained environments. These are highlighted below:

1. Introduced signal processing (IIR) into cryptographic timing defenses.
2. Developed a software-only timing mitigation technique with minimal overhead.
3. Demonstrated real-world applicability in simulated MANET environments.
4. Outperformed existing countermeasures in terms of scalability and efficiency.

References

[1]. Kocher, P. (1996). Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. Advances in Cryptology–CRYPTO'96, 104–113. https://doi.org/10.1007/3-540-68697-5_9

[2]. Bernstein, D. J. (2005). Cache-timing attacks on AES. Technical report, University of Illinois at Chicago.

[3]. Akinboboye, A. J., Oluwole, A. S., Akinsanmi, O., & Oluwafemi, I. B. (2022). Cryptographic algorithms for IoT privacy: A technical review. International Journal of Engineering Trends and Technology, 70(8). https://doi.org/10.14445/22315381/IJETT-V70I8P219

[4]. Brumley, D., & Boneh, D. (2003). Remote timing attacks are practical. Computer Networks, 48(5), 701–716. https://doi.org/10.1016/j.comnet.2005.01.010

[5]. Daemen, J., & Rijmen, V. (2002). The design of Rijndael: AES—the advanced encryption standard. Springer.

[6]. Dinh, T. T. A., Saxena, P., & Shan, Y. (2015). Opaque: An efficient and secure oblivious RAM. ACM SIGSAC Conference on Computer and Communications Security (CCS), 475–488.

[7]. Goodspeed, T. (2014). Introducing new side-channel attacks on embedded systems. Black Hat USA.

[8]. Mangard, S., Oswald, E., & Popp, T. (2007). Power analysis attacks: Revealing the secrets of smart cards. Springer.

[9]. Langley, A. (2010). TLS: Timing attack mitigation. Google Security Blog. https://www.imperialviolet.org/2010/10/30/rounding.html

[10]. Spreitzer, R., & Plos, T. (2013). Cache-access pattern attack on AES. International Workshop on Constructive Side-Channel Analysis and Secure Design, 95–103.

[11]. Gandolfi, K., Mourtel, C., & Olivier, F. (2001). Electromagnetic analysis: Concrete results. Cryptographic Hardware and Embedded Systems (CHES), 251–261. https://doi.org/10.1007/3-540-44709-1_21

[12]. Gierlichs, B., Batina, L., & Preneel, B. (2008). Mutual information analysis: A generic side-channel distinguisher. Cryptographic Hardware and Embedded Systems (CHES), 426–442.

[13]. Sunar, B., Koç, Ç. K., & Stinson, D. R. (2005). Efficient masking for AES. International Workshop on Cryptographic Hardware and Embedded Systems, 209–217.

[14]. Liu, Y., Jin, H., & Zheng, W. (2016). Secure and efficient cryptographic algorithms for embedded systems. Journal of Systems Architecture, 65, 49–62.

[15]. Zhang, Z., Wang, L., & Liu, J. (2020). Lightweight cryptographic solutions for IoT. Sensors, 20(13), 3735.

[16]. Clavier, C., Goubin, L., & Gouget, A. (2000). Enhanced differential power analysis. Advanced Encryption Standard Candidate Conference.

[17]. Suárez-Albela, M., Fraga-Lamas, P., Dapena, A., Fernández-Caramés, T. M., & González-López, M. (2018). Evaluation of lightweight cryptographic algorithms for IoT. Sensors, 18(2), 520. https://doi.org/10.3390/s18020520

[18]. Tiri, K., & Verbauwhede, I. (2005). A logic level design methodology for a secure DPA resistant ASIC. Design, Automation and Test in Europe Conference and Exhibition, 246–251.

[19]. Liu, H., Tang, Y., & Chen, Y. (2020). Adaptive timing equalization for cryptographic processes. IEEE Transactions on Computers, 69(12), 1722–1733.

[20]. Bhattacharya, S., Reith, S., & Schmidt, J. M. (2019). Machine learning and side-channel analysis: An overview. IEEE Transactions on Emerging Topics in Computational Intelligence, 3(4), 327–339.

[21]. Wang, Y., Zhou, H., & Wang, Z. (2018). Power-efficient side-channel attack mitigation strategies. IEEE Access, 6, 28994–29005.

[22]. Alam, S., Singh, S., & Gupta, B. B. (2022). A review on edge computing security using side-channel-resistant cryptographic primitives. Computer Communications, 192, 54–66.

[23]. Zhang, L., & Zhang, Y. (2019). Design of secure cryptographic filters. Journal of Cryptographic Engineering, 9, 205–216.

[24]. Kumar, R., & Singh, N. (2022). A comprehensive survey on MANET security architecture. Wireless Personal Communications, 124(2), 1209–1235.

[25]. Barenghi, A., Breveglieri, L., Koren, I., & Naccache, D. (2012). Fault injection attacks on cryptographic processors. IEEE Design & Test of Computers, 29(3), 58–66.

[26]. Haider, R., Kamal, A., & Ahmed, S. H. (2019). Enhancing AES security using masking techniques for IoT. International Journal of Electronics and Communications, 105, 1–10.

[27]. Tian, X., & Wang, H. (2022). Energy-aware cryptographic design for mobile sensor networks. Journal of Communications and Networks, 24(1), 13–23.

[28]. Alomari, E., Al-Bahadili, H., & Bani-Salameh, H. (2022). A framework for secure routing in MANETs. Ad Hoc Networks, 125, 102715.

[29]. Benadjila, R., et al. (2018). Security analysis and countermeasures of AES implementations. Journal of Information Security and Applications, 43, 55–67.

[30]. Das, A., Deb, B., & Banerjee, A. (2021). Lightweight constant-time cryptographic operations for IoT. Microprocessors and Microsystems, 83, 103985.

[31]. Mukherjee, P., & Kar, S. (2020). A comparative study of side-channel attack defenses in embedded systems. ACM Computing Surveys, 53(6), 1–32.

[32]. Okafor, M., Adebayo, O., & Ajayi, B. (2023). Signal processing approaches for mitigating side-channel attacks. Journal of Cybersecurity and Privacy, 3(2), 157–175.

[33]. Seifert, J. P. (2005). Lattice attacks on cryptographic primitives via timing analysis. IACR Cryptology ePrint Archive, Report 2005/187.

[34]. Naccache, D., & M'Raïhi, D. (2001). How to measure timing information leakage. Journal of Cryptographic Engineering, 1(2), 101–114.

[35]. Pradhan, S., & Barik, R. (2023). Comparative analysis of AES timing stabilization filters. International Journal of Computer Applications, 182(25), 20–28.

[36]. Hutter, M., & Schmidt, J. M. (2013). The temperature side channel and its security implications. ACM Transactions on Embedded Computing Systems, 13(1), 1–28.

[37]. Zhang, L., & Wu, M. (2018). Efficient and secure AES in constrained devices. International Journal of Network Security, 20(4), 663–672.

[38]. Guo, C., & Liu, H. (2020). Optimizing noise filtering in real-time encryption. IEEE Transactions on Dependable and Secure Computing, 17(5), 1132–1145.

[39]. Bhunia, S., & Tehranipoor, M. (2018). Hardware security: A hands-on learning approach. Morgan Kaufmann.

[40]. Trichina, E., Korkikyan, A., & Hamburg, M. (2005). Masking techniques in AES software implementations. CHES, 92–102.

[41]. Sun, Y., Yu, H., & Guo, X. (2023). A machine learning-enhanced IIR filtering model for AES timing defense. IEEE Internet of Things Journal, 10(3), 1832–1845.