TPKEY: Using TPMS Signals for Secure and Usable Intra-Vehicle Device Authentication

Omar Achkar*, Larry Nissen*, Shahryar Raza*, Rushikesh Shirsat†, Neil Klingensmith†,
George Zouridakis*, and Kyu In Lee*
*Department of Information Science Technology, University of Houston, Houston, Texas 77204
†Department of Computer Science, Loyola University Chicago, Chicago, Illinois 60660
Email: *{oachkar, ljnissen, saraza3, gzourida, klee48}@cougarnet.uh.edu, †{rshirsat, nklingensmith}@luc.edu

2025 IEEE International Conference on Cyber Security and Resilience (CSR) | 979-8-3315-3591-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/CSR64739.2025.11130079

***Abstract*—Modern vehicles increasingly support wireless connectivity with various mobile devices, but existing authentication methods like Bluetooth pairing create unnecessary friction in shared mobility scenarios. This paper presents TPKEY, a novel zero-involvement authentication protocol that leverages tire pressure monitoring system's (TPMS) radio transmissions to allow secure and autonomous device authentication within vehicles. The system extracts high-entropy keys from demodulated TPMS signal characteristics that are only accessible within the vehicle's physical boundary. Our implementation addresses key technical challenges, including precise time synchronization between devices and efficient entropy extraction from raw signals. Extensive evaluation of TPKEY achieves 100% authentication success rate for legitimate devices, while maintaining 0% false acceptance rate for adversaries, with key generation time of 1.3 s. Furthermore, TPKEY maintains reliable performance across various driving conditions and device positions within the vehicle, offering a practical solution for seamless and secure device authentication in modern automotive environments.**

***Index Terms*—Tire pressure monitoring system (TPMS), Zero-interaction authentication, Device authentication**

I. Introduction

Modern automotive computer systems have evolved into sophisticated platforms that integrate wireless connectivity with a wide array of devices, including smartphones, tablets, wearables, and fleet trackers. While Bluetooth has emerged as the predominant standard for connecting mobile devices to automotive computers, its limitations become apparent in the context of shared mobility. Services like Zipcar [1] and robotaxi platforms must manage a constant stream of different devices connecting to their in-vehicle infotainment (IVI) systems during brief trips. This paradigm calls for more agile authentication procedures capable of rapid device authentication and automatic credential management. Current approach of requiring manual PIN or password entry for authenticating devices creates unnecessary friction in the user experience and fails to meet the dynamic needs of shared vehicle environments. Furthermore, the need to periodically remove inactive authentication keys (i.e., when the rider leaves the vehicle) from the vehicle creates additional maintenance overhead and degrades the user experience. Unlike personal vehicles where pairing is a one-time setup, shared mobility services require continuous authentication management as multiple users connect and disconnect their devices throughout each day.

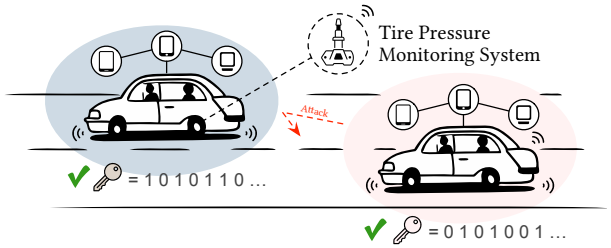

Fig. 1. Devices inside left vehicle generate identical authentication key based on TPMS signal, while devices in nearby vehicle generate different key.

Recently, zero-involvement authentication (ZIA) has emerged as a compelling solution to provide fast, usable, and secure authentication for various wirelessly-connected systems. Rather than relying on traditional key sharing methods like passwords and PINs, ZIA systems autonomously generate authentication keys from environmental context shared by nearby devices. The key advantage of ZIA systems lies in their ability to seamlessly adapt to real-world conditions without requiring user intervention. By continuously monitoring and analyzing environmental signals within a confined physical space, ZIA systems can dynamically generate and revoke authentication keys as devices naturally move in and out of the network. In vehicular domain, previous research has explored various contextual signals, such as ambient audio [2], acceleration [3], and shared visual patterns [4]. However, the inherent challenges of vehicular environments, characterized by constant noise and lack of high entropy signal, have posed significant obstacles for such methods.

This work introduces TPKEY, an authentication mechanism that builds secure authentication keys from a vehicle's tire pressure monitoring system's (TPMS) wireless transmissions. By leveraging readily available TPMS transmissions, devices inside the vehicle capture physical-layer measurements, such as demodulated signal characteristics and message timing patterns from the transmitted TPMS data. While these signals remain unencrypted, the vehicle's physical structure creates unique signal propagation and attenuation characteristics that are inherently difficult for external adversaries to replicate. Effectively, this transforms the vehicle's physical environment into a natural security barrier, as authentic signal profiles can

only be received within the vehicle. Fig. 1 demonstrates the intended usage; TPMS signal is captured by devices within the vehicle, creating a unique signal profile that turns into unique key, which cannot be reproduced by outside adversaries due to the variance in signal quality from a combination of distance and car body interference.

Implementing fast and secure ZIA system for the automotive environment presents several significant challenges. The first challenge is to identify an environmental signal that exhibits three critical characteristics: high entropy, strong correlation among in-vehicle devices, and minimal correlation with external devices. While TPMS radio transmissions effectively meet these criteria, achieving reliable system performance demands precise signal acquisition and processing methodologies. Secondly, time synchronization between authenticating devices introduces another layer of complexity, particularly because typical mobile devices lack precise hardware timing capabilities. TPKEY addresses this through a novel approach where one device shares a carefully selected TPMS signal segment for temporal alignment without compromising the security of the authentication process. The final challenge is extracting adequate entropy from the TPMS signal, which is inherently a low-entropy signal. TPKEY overcomes this limitation by utilizing the raw demodulated TPMS signal rather than binary packet data, accessing substantially higher entropy levels as demonstrated in Section II. The key contributions of this paper are as follows:

- We introduce the novel intra-vehicle device authentication method TPKEY, leveraging TPMS signal characteristics to establish secure authentication keys.
- We present novel solutions to address the practical implementation challenges of TPKEY on commercial devices, focusing on timing misalignment and signal processing techniques.
- We present experimental validation of TPKEY across diverse real-world driving conditions, achieving a 100% authentication success rate for legitimate device pairs while maintaining a 0% success rate for potential adversaries.

II. Background

A. Tire Pressure Monitoring Systems (TPMS)

Under the TREAD Act [5], every new car sold in the United States after 2007 must include a TPMS to reduce accidents caused by tire failures. Such sensor is installed in each tire and periodically transmits data to the vehicle at 315 MHz or 433.92 MHz, providing information about tire pressure along with additional metrics such as temperature, battery status, and a unique sensor identifier [6]. Typically, the transmission occurs about once per minute, or more often upon detecting a sudden pressure drop (e.g., a blowout). Although these signals follow manufacturer-specific proprietary protocols, they are transmitted without encryption and are openly accessible to the public [6].

To investigate the feasibility of extracting unique signal characteristics from TPMS transmissions, we conduct an initial driving experiment using an RTL software defined radio (SDR) [7] connected to a laptop running decoding software [8] tuned to 315 MHz. We collect data while driving inside the vehicle to monitor transmissions from the TPMS sensors. Fig. 2(a) plots temperature and pressure measurements from the front driver's side sensor during a 30-minute highway drive, overlaid with vehicle's speed data reported from the vehicle's on-board diagnostics (OBD)-II port. As expected, the tire temperature shows consistent increase during prolonged high-speed driving, with pressure following a similar but less pronounced trend. These decoded sensor readings proved inadequate for generating secure authentication keys for two critical reasons: 1) the measurements exhibited insufficient entropy. The predictable correlation between vehicle speed and sensor readings resulted in highly deterministic patterns. 2) the sampling frequency are insufficient for practical implementation, yielding only 140 data points per wheel over 30 minutes, making rapid device authentication impractical.

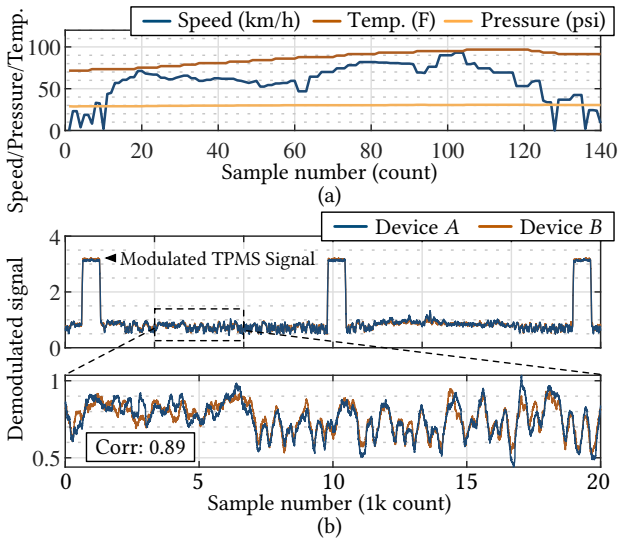

Fig. 2. (a) Speed (km/h), temperature (F) and pressure (psi) measurements from one wheel over the course of a 30-minute drive along the highway. (b) Raw signal captured by two devices within the same car.

On the other hand, examination of the raw demodulated TPMS radio signals reveals more promising characteristics. We position two SDRs inside the vehicle and capture raw demodulated signals. The results, illustrated in Fig. 2(b), show distinct TPMS transmission peaks interspersed with baseline background noise. The zoomed-in analysis of this background noise reveals both significant temporal randomness and strong inter-device correlation, with a correlation coefficient of 0.89 between devices. Moreover, the noise characteristics appear independent of the deterministic measurements shown in Fig. 2(a). This combination of unpredictability and fluctuating characteristics, on top of consistent correlation between independent receivers form the foundation of TPKEY's authentication mechanism: the background noise provides approximately 500,000 samples of high-entropy, correlated data per second, allowing generation of robust cryptographic keys from these shared environmental characteristics.

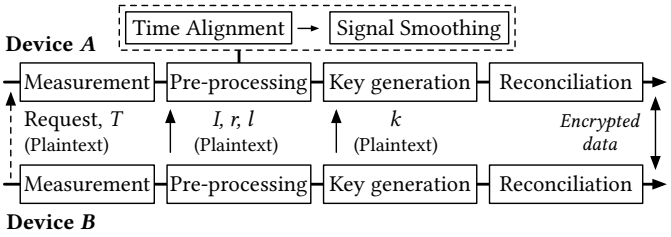

Fig. 3. TPKEY's four stage authentication protocol to extract identical keys on two devices, A and B.

B. System and Threat Models

TPKEY considers a standard automotive environment where devices within the vehicle seek authenticate each other to establish high-bandwidth wireless connections, such as Bluetooth or Wi-Fi, by generating identical symmetric keys. These devices may be passenger-owned devices or components of the vehicle's built-in infotainment system. We assume all devices are equipped with RF hardware or SDR capable of monitoring TPMS frequencies (315 MHz or 433 MHz).

Our threat model makes a distinction between two classes of mobile devices: *legitimate devices* are physically inside the car and *adversarial devices* are located outside the car. Adversaries may be stationary (e.g., pedestrians on the roadside) or mobile in a nearby vehicle. Based on the vehicle model, the attacker has knowledge of which frequencies the target's TPMS utilizes. The adversary's objective is to establish unauthorized authentication with either the vehicle or legitimate mobile devices inside the victim vehicle to compromise system's control and integrity. We assume that adversaries can intercept and monitor any unencrypted wireless communications between legitimate devices during the authentication process and can also attempt to authenticate with the vehicle or legitimate devices by following the TPKEY protocol. We consider denial of service attacks, such as jamming, to be outside the scope of TPKEY as such attacks are applicable to any wireless communication mechanism and are not specific to our protocol. This aligns with threat models in similar ZIA systems presented in [3], [9], [4].

III. Proposed Authentication Protocol

This section presents the overall TPKEY protocol, which allows devices to establish secure communication channels without prior shared knowledge, as illustrated in Fig. 3. The protocol consists of four stages: Measurement, Pre-processing, Key Generation and Reconciliation.

Measurement: The protocol begins when Device B transmits authentication request message containing parameter T (plaintext), which defines the required number of signal samples to capture. Upon receiving the request, both devices initiate independent capture of raw demodulated signals, each collecting T samples. However, the inherent transmission delay of the request message creates temporal misalignment between the captured signals, manifesting as d-sample displacement between the measurements, as illustrated in Fig. 4(a). This misalignment prevents direct use of the signals for symmetric key generation without performing signal alignment.

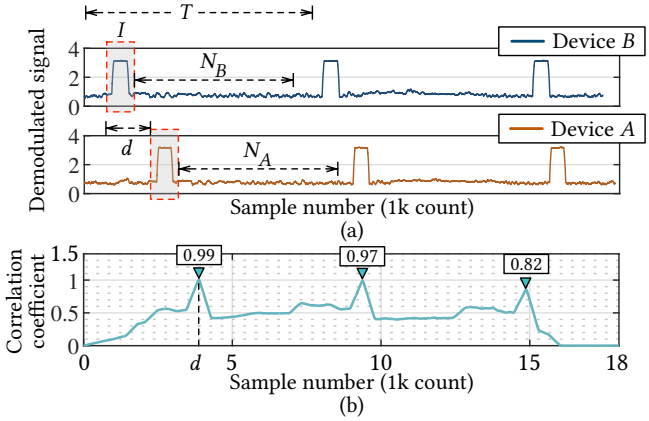

Fig. 4. (a) Pre-processing stage to align two signal with offset d. Devices exchange A send reference point I to align signal. (b) Delay offset d can be found by cross correlating I with device B's own signal.

Pre-processing: The pre-processing stage primarily serves two functions in our protocol: establishing precise temporal alignment between devices and extracting secure noise portions suitable for key generation. This stage begins with temporal alignment, where Device B analyzes its captured signal buffer to identify the first TPMS modulated signal using peak detection method. This identified segment serves as an initialization reference buffer I, which Device B transmits to Device A in plaintext to establish a common reference point. Upon receiving I, Device A utilizes a sliding window correlation approach to locate the corresponding signal segment in its own buffer, as illustrated in Fig. 4(b). This allows precise determination of the temporal displacement d, even when multiple TPMS peaks are present. The correlation analysis demonstrates highest similarity at the first peak with a coefficient of 0.99, successfully identifying the correct temporal offset despite other peaks showing strong correlations.

Since the initialization reference buffer I is transmitted over a public channel, it cannot be incorporated into the key generation process without compromising security. To address this, Device B transmits an additional parameter l that specifies the exact number of samples to be used after the aligned reference point I. Both devices then extract their respective noise signals (N_A and N_B), each comprising precisely l samples following their aligned I segments. Since these extracted noise segments are derived solely from locally captured signals and never transmitted over public channels, they maintain their secrecy for secure key generation.

The final pre-processing step enhances signal quality through root mean square (RMS) filtering applied independently by both devices to their respective noise signals (N_A and N_B). By computing the RMS value, r, within each window position, the filter effectively normalizes signal variations and eliminates transient peaks that may occur in only one device's buffer. This ensures signal consistency between devices, improving the likelihood of generating matching keys.

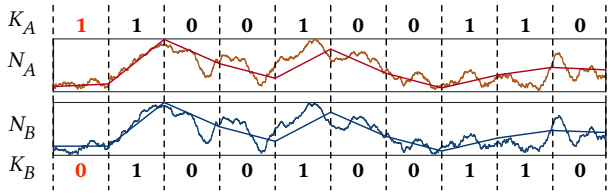

Fig. 5. Key generation stage where Device A and Device B independently extract bit sequences (K_A and K_B) from their respective noise signals (N_A and N_B).

Key Generation: Following pre-processing, each device owns time-aligned and highly correlated noise segments N_A and N_B. Device B initiates key generation by transmitting parameter k (plaintext), which specifies the desired key length in bits. Both devices then segment their respective noise segments into k equal-width bins, with each bin contributing one bit to the final key through slope-based quantization. Within each bin, the system applies least squares linear interpolation to determine the best-fit line slope: positive slopes generate a bit `1` while negative slopes produce `0`. As illustrated in Fig. 5, this quantization method produces nearly identical bit sequences K_A and K_B on both devices. While the noise signals N_A and N_B show strong correlation, minor local fluctuations or slight misalignment can occasionally result in bit disagreements, particularly when the calculated slopes are close to zero or when the signal exhibits increased noise.

Reconciliation: To properly authenticate two devices using symmetric key cryptography, the generated bit sequences (K_A and K_B) must be identical across both devices. While the initial key generation process typically achieves high similarity, signal variations can introduce occasional bit discrepancies, as shown in the first bit of Fig. 5 where flattened signals lead to disagreement. To address these mismatches without compromising security, TPKEY implements a widely known key reconciliation protocol based on error-correcting codes (ECC) [10]. Such reconciliation process segments the extracted bit sequences and maps each segment to corresponding precomputed codewords, ensuring no exposure of the actual key bits. Specifically, when using a Hamming(n, k) code, each k-bit segment is transformed into an n-bit codeword that minimizes the Hamming distance from the original sequence. This error correction mechanism only functions effectively when the initial bit error rate remains below a predetermined threshold, ultimately achieving the perfect bit agreement required for secure symmetric key operations.

IV. Implementation and Evaluation

In this section, we present a comprehensive evaluation of TPKEY under realistic usage scenarios and adversarial conditions. Our evaluation comprises controlled and real-world deployment scenarios, providing thorough assessment of the TPKEY.

A. Experimental Setup and Metrics

The signal acquisition setup employs RTL-SDR V4 software-defined radios (each representing a single device)

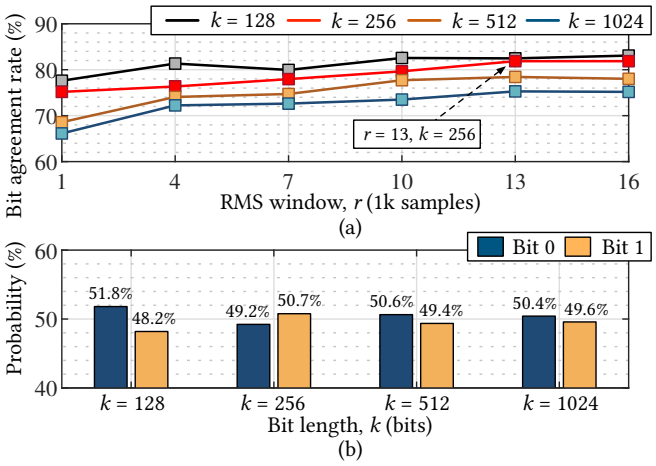

Fig. 6. (a) Mean BAR between two devices across different RMS window sizes (r) and key lengths (k). (b) Probability of generating bits (0 or 1) for varying key lengths(k) with $r = 13k$.

connected to a Linux-based laptop running RTL-433 software to capture and demodulate TPMS signals. These signals are then processed using MATLAB scripts that implement the four-stage protocol detailed in Section III. Evaluations were carried out under various environmental conditions, including city, rural, and highway driving scenarios, with 2019 Toyota RAV4 serving as the test vehicle.

We focus on two main evaluation metrics: the *bit agreement rate (BAR)* and the authentication *success rate (SR)*. The bit agreement rate measures the fraction of identical bits in two generated keys before the reconciliation stage. The success rate reflects the proportion of authentication attempts that achieve 100% bit agreement rate after reconciliation that can be used to generate symmetric key.

B. Parameter Selection

The performance of TPKEY is primarily governed by two parameters: the RMS window size (r) for signal smoothing and the key length (k) for key generation. To identify optimal configuration, we evaluate the bit agreement rates between two devices installed within a vehicle: one on the driver's side and the other on the passenger's side. We collect noise signals (N) with length (l) of 1 million samples over a 2 second interval, while varying r from 1 to 16 kilosamples and k of 128, 256, 512, and 1024 bits. As illustrated in Fig. 6(a), the mean BAR shows a positive correlation with increasing r across all key lengths. This stems from the fact that small RMS window sizes fail to adequately smooth the signal, resulting in excessive noise that results in inconsistent bit generation between devices. Notably, the increase in BAR gradually plateaus beyond 13 kilosamples, indicating diminishing returns with further increase.

The selection of k also involves a trade-off. Extracting too many bits results in overly granular slope comparisons that lower the BAR, while too few keys impacts the TPKEY's security characteristics. As illustrated in Fig. 6(b), when the key length is too short ($k = 128$), the system fails to adequately capture the signal's fluctuations, resulting in

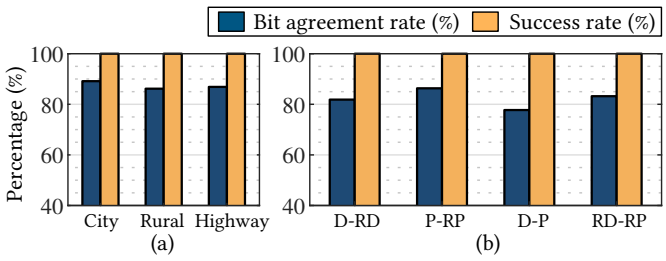

Fig. 7. Mean BAR and SR for different (a) driving environment and (b) location-pair within the vehicle (Driver (D), Passenger (P), Rear Passenger (RP), and Rear Driver (RD)).

an uneven distribution of bits as shown in with bit `0` at 51.8% and bit `1` at 48.2%. This bias can potentially help an adversary in predicting the generated key. In contrast, length of 256 yields a more balanced distribution (49.2% for bit `0` and 50.7% for bit `1`), enhancing the security of the system. Based on our analysis, we selected $k = 256$ with r of 13 kilosamples as the optimal configuration, striking effective balance between security and performance. This parameter combination achieves mean BAR above 80%, which allows us to empirically choose key reconciliation threshold of 75% for subsequent evaluations.

C. Driving environment & Device Location

Next, we evaluate the performance of TPKEY in various real-world driving environments and device placements, with over 100 sets of generated 256 bit keys. Figure 7(a) illustrates the mean BAR and SR across three distinct environments: city, rural, and highway. In the city setting with high-traffic areas, our system achieves the highest mean BAR of 89.1%. The rural and highway environments yields slightly lower BARs of 86.1% and 86.8%, respectively. Nevertheless, all environments maintains mean BAR above 85% even at highway speeds and under varying RF conditions. More importantly, TPKEY achieves 100% authentication success rate in all environments we tested. This demonstrates TPKEY's reliability across diverse real-world scenarios.

The impact of device positioning within the vehicle was evaluated through four device location pairs, as illustrated in Fig. 7(b): Driver-Rear Driver (D-RD), Passenger-Rear Passenger (P-RP), Driver-Passenger (D-P), and Rear Driver-Rear Passenger (RD-RP). The P-RP configuration exhibits the highest mean BAR at 86.3%, while other configurations maintained rates above 79% with D-RD, D-P, and RD-RP showing 81.6%, 79.2%, and 83.2%, respectively. Notably, success rates remained 100% across all placement combinations, demonstrating that TPKEY's effectiveness in signal reception regardless of different device positioning.

D. Adversarial scenarios

We next evaluate TPKEY's security under adversarial scenario. In our setup, two legitimate devices were positioned inside the vehicle while adversarial device is positioned 1 m outside. All devices were configured identically, and the

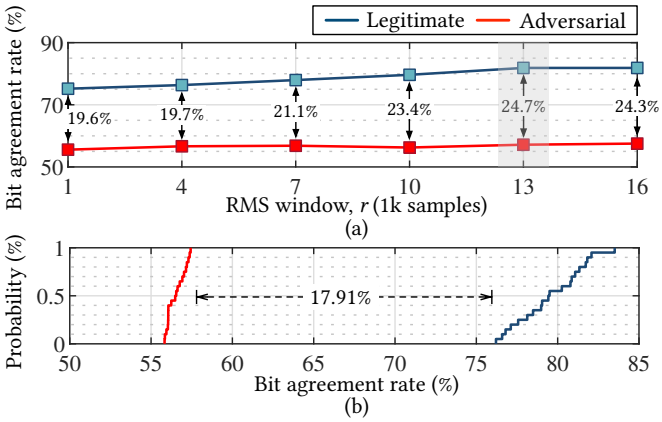

Fig. 8. (a) BAR between legitimate and adversarial devices across different RMS window sizes. (b) Probability distribution BARs between legitimate and adversarial devices.

adversarial device was set to request authentication to in-vehicle devices using same messages and protocols as those employed by legitimate ones. To additionally maximize the attackers success rate, we perform the authentication in a stationary environment with minimal RF interference and we only activate the driver's side TPMS sensor. Fig. 8(a) illustrates how varying the RMS window size affects mean BARs for both legitimate devices and the external adversary. Legitimate devices maintained rates between 75% and 85% across RMS window sizes ranging from 1 to 16 kSamples. In contrast, the adversarial device's achieved relatively low BAR, with rates consistently between 50% and 55%, essentially no better than rate of random guess.

Fig. 8(b) provides a probabilistic analysis of BARs, highlighting the clear separation between legitimate and adversarial performance with parameter of $k = 256$ and $r = 13$. The data reveals a significant security margin, with a 17.91% gap between the highest adversarial BAR and the lowest BAR. This substantial separation demonstrates TPKEY's robust security characteristics—legitimate devices consistently achieved bit agreement rates between 75–85%, while adversarial attempts never exceeded 60%. This clear delineation ensures that TPKEY can reliably authenticate legitimate devices while effectively rejecting unauthorized access attempts.

V. Discussion

In this section, we highlight few important considerations for real-world deployment: authentication time, security boundaries, practicality and application.

Authentication Time: TPKEY demonstrates efficient key generation performance, producing k bit keys from a single second of TPMS noise signal. After taking account 75% reconciliation threshold, which effectively reduces the entropy of the final key, the system requires 5.2 ms per effective bit in the resulting key. As shown in Fig. 9(a), this translates to total generation times of 0.6 s and 1.3 s for keys with effective entropy equivalent to 128 and 256 bit keys, respectively. Prior study on Bluetooth authentication shows that users typically spend 27 s to manually copy and confirm an 8-digit PIN [11].

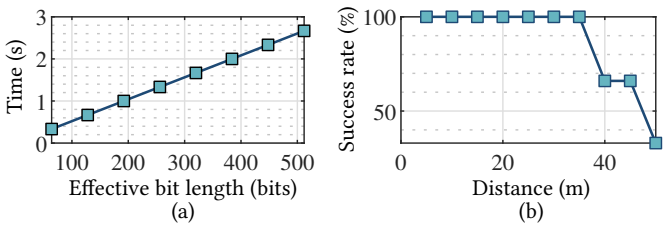

Fig. 9. (a) Effective entropy and (b) Communication success rate vs. Distance.

In comparison, TPKEY offers two significant advantages: it can generate longer keys within significantly less time while providing fully automated authentication.

Physical Security Boundaries: The propagation characteristic of TPMS signals inherently establishes a physical security perimeter around the vehicle. As shown in Fig. 9(b), the communication success rate between TPMS transmitter and receiver remains at 100% up to approximately 35 m, after which it drops to below 50% by 45 meters. This degradation creates a well-defined spatial boundary for secure communication.

Practicality and Application: The transition from prototype to wide-scale adoption presents important challenges. The current implementation requires hardware capable of receiving TPMS transmissions, which may be impractical for widespread consumer deployment. Integration with existing mobile devices would require either adding TPMS-compatible receivers or modifying existing wireless hardware. Future implementations could potentially leverage existing wireless chips in mobile devices through firmware updates to enable SDR capabilities, eliminating the need for additional hardware. Beyond intra-vehicle device authentication, TPKEY has potential applications in vehicle diagnostics and maintenance. It can provide technicians with authenticated access to onboard diagnostic systems and the controller area network bus while maintaining security boundaries that prevent unauthorized access to these critical vehicle systems.

VI. Related work

ZIA have been extensively explored for securing wireless communications between IoT devices. Early work has demonstrated the feasibility of using ambient audio for device authentication in general environments [12], [13]. This concept was further developed to examine various contextual signals including luminosity [14], RF characteristics [15], power line noise [16], timing characteristics [17] and visual channels [18]. However, these approaches rely on context signals that are often unavailable or has low-entropy in vehicular settings, as cars are designed to isolate the cabin from external light, sound, and electromagnetic interference.

In the vehicular domain, prior work has mainly leveraged visual sensing [4] or inertial measurement unit (IMU) data by capturing shared road conditions and vehicle vibration patterns [3], [9], [19], [2]. While these approaches demonstrate the feasibility of using vehicular motion for authentication, they suffer from limited entropy in IMU signals and require long sensing periods to generate keys. In contrast, TPKEY leverages readily available TPMS signals that provide high entropy while requiring minimal sensing time, addressing the key limitations of previous intra-vehicle authentication systems.

VII. Conclusion

This paper presents TPKEY, a novel protocol that allows secure and usable device authentication in vehicles by leveraging TPMS signal characteristics. Our approach addresses the limitations of existing methods by eliminating manual interaction while maintaining strong security guarantees. Through comprehensive evaluation, we demonstrate that TPKEY achieves reliable authentication across diverse real-world conditions while effectively preventing unauthorized access. The system successfully generates cryptographic keys with high entropy (256 bits) in under 1.3 s, representing a promising step toward seamless and secure device authentication in modern vehicles.

Acknowledgements

This work was supported by the US Department of Transportation (USDOT) Tier 1 University Transportation Center (UTC) Transportation Cybersecurity Center for Advanced Research and Education (CYBER-CARE) (Grant No. 69A3552348332); and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00463802).

References

[1] "Zipcar," https://www.zipcar.com.
[2] M. Fomichev *et al.*, "Perils of zero-interaction security in the internet of things," *ACM IMWUT 3*, 2019.
[3] J. Han *et al.*, "Convoy: Physical context verification for vehicle platoon admission," in *ACM HotMobile '17*.
[4] J. Veselsky *et al.*, "Establishing trust in vehicle-to-vehicle coordination: A sensor fusion approach," in *DI-CPS '22*.
[5] "Tread act," https://www.congress.gov/bill/106th-congress/house-bill/5164.
[6] I. Rouf *et al.*, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *USENIX Security '10*.
[7] "Rtl-sdr.com," https://www.rtl-sdr.com.
[8] C. W. Zuckschwert, "RTL 433," https://github.com/merbanan/rtl_433.
[9] K. Lee *et al.*, "ivpair: Context-based fast intra-vehicle device pairing for secure wireless connectivity," in *ACM WiSec '20*.
[10] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *ACM CCS '99*.
[11] E. Uzun *et al.*, "Usability analysis of secure pairing methods," *Financial Cryptography and Data Security*, 2007.
[12] D. Schurmann and S. Sigg, "Secure communication based on ambient audio," *IEEE TMC*, 2013.
[13] M. Miettinen *et al.*, "Revisiting context-based authentication in iot," in *ACM DAC '18*.
[14] M. Miettinen *et al.*, "Context-based zero-interaction pairing and key evolution for advanced personal devices," in *ACM CCS '14*.
[15] S. Mathur *et al.*, "Proximate: Proximity-based secure pairing using ambient wireless signals," in *ACM MobiSys '11*.
[16] K. Lee *et al.*, "Voltkey: Continuous secret key generation based on power line noise for zero-involvement pairing and authentication," *ACM IMWUT 3*, 2019.
[17] J. Han *et al.*, "Do you feel what i hear? enabling autonomous iot device pairing using different sensor types," in *IEEE S&P '18*.
[18] N. Saxena *et al.*, "Secure device pairing based on a visual channel (short paper)," in *IEEE S&P '06*.
[19] M. Fomichev *et al.*, "Fastzip: faster and more secure zero-interaction pairing," in *ACM MobiSys '21*.