VWATTACKER: A Systematic Security Testing Framework for Voice over WiFi User Equipments

Imtiaz Karim[†,*], Hyunwoo Lee[§,*], Hassan Asghar[§], Kazi Samin Mubasshir[‡],
Seulgi Han[§], Mashroor Hasan Bhuiyan[†], Elisa Bertino[‡]
[†]The University of Texas at Dallas, [§]Korea Institute of Energy Technology, [‡]Purdue University
[†]{imtiaz.karim,mashroorhasan.bhuiyan}@utdallas.edu, [§]{hwlee,hassanasghar,ab9938}@kentech.ac.kr
[‡]{kmubasash,bertino}@purdue.edu

***Abstract*—We present VWATTACKER, the *first systematic testing* framework for analyzing the security of Voice over WiFi (VoWiFi) User Equipment (UE) implementations. VWATTACKER includes a complete VoWiFi network testbed that communicates with Commercial-Off-The-Shelf (COTS) UEs based on a simple interface to test the behavior of diverse VoWiFi UE implementations; uses property-guided adversarial testing to uncover security issues in different UEs systematically. To reduce manual effort in extracting and testing properties, we introduce an LLM-based, semi-automatic, and scalable approach for property extraction and testcase (TC) generation. These TCs are systematically mutated by two domain-specific transformations. Furthermore, we introduce two deterministic oracles to detect property violations automatically. Coupled with these techniques, VWATTACKER extracts 63 properties from 11 specifications, evaluates 1,116 testcases, and detects 10 issues in 21 UEs. The issues range from enforcing a zero DH key to supporting weak algorithms. These issues result in attacks that expose the victim UE's identity or establish weak channels, thus severely hampering the security of cellular networks. We responsibly disclose the findings to all the related vendors. One of the vulnerabilities has been acknowledged as CVE-2025-20667.**

I. INTRODUCTION

Voice over WiFi (VoWiFi) is a mobile call service that aims to extend network coverage by utilizing WiFi in areas where a cellular network is unavailable or the cellular signal is weak. Poor signal is not a new issue because typically areas within buildings or underground areas are not covered by conventional mobile networks [2], [35]. Due to these limitations of the current cellular network infrastructure, VoWiFi usage is increasing with the large number of WiFi networks deployed in homes/offices and the growth of public WiFi [7]. The VoWiFi market is estimated to reach USD 21.99 billion in 2030 from USD 9.59 billion expected in 2025 with an annual growth rate of more than 18% [1]. Such widespread use of VoWiFi requires robust security guarantees.

Prior research and scope. Although previous works have analyzed the security of the VoWiFi protocol specifications [6], [21], [25], [36], [41], VoLTE specifications and implementations [19], [24], and cellular network implementations in general [10], [14], [20], [31], [39], there is no systematic framework for analyzing VoWiFi implementations with a complete open-source VoWiFi commercial UE testbed (see Table I). Furthermore, the main body of work [6], [25], [41] on VoWiFi looks at scattered parts of the protocol, and the approaches are ad hoc and manual. Lee *et al.* [21] verify the complete protocol based on formal verification, but their analysis focuses only on the specifications. Concerning the security analysis of VoWiFi implementations, our only reference is the conformance test suites [3] provided by 3GPP. The conformance test suites aim to ensure that VoWiFi implementations meet the minimum requirements. However, these testcases (TCs) have limitations. They (i) focus on functional requirements rather than security, and (ii) do not evaluate the behaviors of VoWiFi implementations in adversarial scenarios. Although there has been a broad body of work on cellular implementation testing [10], [20], [31], those frameworks are built on top of different open-source cellular network implementations and are not general enough to include VoWiFi testing.

Motivated by this significant gap in the testing of VoWiFi implementations, we develop the first *systematic* security testing framework–VWATTACKER to evaluate the security of VoWiFi UE implementations. At a high level, VWATTACKER is an adversarial property-guided testing framework. However, we include significant innovations to reduce manual effort and substantially increase the scope of testing.

Challenges. The first challenge for the adversarial testing on COTS UEs is that there is no open-source VoWiFi testbed that controls and synchronizes the many entities involved in VoWiFi communications (e.g., UEs, the ePDG, and the IMS). The second challenge is related to the scalability of property-guided adversarial testing. In this broader challenge, we face two sub-challenges. First, such a testing technique requires significant manual effort to extract properties from the specifications, and this limits comprehensive security testing [20], [31]. Second, it is difficult to detect when a property violation has occurred without any manual intervention. As the specifications are written in natural language and contain multiple underspecifications [21], it is not always possible to infer the correct behavior.

Approach. To address the first challenge and bridge the gap for testing VoWiFi UE implementations, VWATTACKER includes a complete VoWiFi network testbed that we built based on open-source software (*StrongSwan* [37] for ePDG, *Kamailio* [16] for IMS, and *FHoSS* [13] for Home Subscriber Server (HSS)). The testbed is currently developed for 4G LTE because, with regards to VoWiFi, 4G LTE is still the most dominant technology [12]. To control VoWiFi entities, we design a *command-report*

* Imtiaz Karim and Hyunwoo Lee contributed equally to this research.

979-8-3315-4961-9/26/$31.00 ©2026 IEEE

Paper	Systematic Frame-work	Implementation Analysis	Complete VoWiFi UE Testbed	Scalable Property Extraction
Wi-Not-Calling [6]		✓		
Xie *et al.* [41]		✓		
Lu *et al.* [25]		✓		
Shi *et al.* [36]		✓		
Gegenhuber *et al.* [11]		✓		
VWANALYZER [21]	✓			
Li *et al.* [24]	✓			
Kim *et al.* [19]	✓			
DIKEUE [14]	✓	✓		
5GBaseChecker [39]	✓	✓		
LTEFuzz [20]	✓	✓		
DoLTEst [31]	✓	✓		
VWATTACKER	✓	✓	✓	✓

TABLE I: Comparison of VWATTACKER with other analyses on cellular networks, VoWiFi, and VoLTE

protocol based on a central-peripheral paradigm, where a central controller manages communication between all entities. We also implement a simple JSON-based interface that allows testers to describe any sequence of messages containing specific attribute values, enabling us to conduct efficient adversarial testing.

In a recent independent and concurrent project, Osmocom has developed an open-source *osmo-ePDG* for VoWiFi UE connectivity [30]. Such an effort not only attests to the importance of the problem but also motivates its urgency. Though promising, osmo-ePDG has several key differences from our testbed. First, osmo-ePDG is a testbed just for VoWiFi UE connectivity and not for security testing. In contrast, our testbed is designed for security testing, featuring a JSON-based testing format for describing adversarial and out-of-order messages to be sent to UEs, as well as a separate control architecture for facilitating easier testing. Second, osmo-ePDG is a module extending the osmocom mobile network, while ours is dedicated to the VoWiFi security analysis with minimal additional functions.

To address the second challenge and enhance property-guided testing, we design TESTGEN that adopts a scalable and semi-automatic Large Language Model (LLM)-based property extraction approach to extract diverse properties. We leverage the *in-context learning* capability of the LLMs to learn from the VoWiFi specifications and generate properties using Retrieval Augmented Generation (RAG) [23]. With just a few example properties and the VoWiFi specifications as context, an LLM can accurately generate additional properties and encode them as Primary Test Cases (PTCs), eliminating costly manual property extraction. These PTCs are then mutated using the two transformations, creating Adversarial Test Cases (ATCs) and sent to the testbed. As messages that contain attributes are exchanged in the VoWiFi protocol, we design two types of transformations: (i) *message-level* transformation inserts, replaces, drops, or replays a target *message*; and (ii) *attribute-level* transformation inserts, updates, or drops an *attribute* in a target message. To automatically detect property violations, we design two different oracles-*function* and *liveness* oracles, where the former detects invalid functions of UEs, and the latter identifies deadlock states of UEs.

Results. With VWATTACKER, we evaluate 21 UEs from 14 different UE vendors and 5 baseband vendors with Android versions ranging from 7 to 14. For the property testing, TESTGEN extracts 63 PTCs from 11 VoWiFi-related specifications. After applying two types of transformations, we get 1,116 ATCs. We report 10 unique issues. Notable among them are issues where implementations: (i) support and accept weak/deprecated algorithms (e.g., DES or MD5); and (ii) accept packets without a nonce or a DH key that can be used to bypass the DH key security to expose a device's IMSI. The impact of these attacks ranges from privacy leaks to establishing weak channels with weak encryption and integrity algorithms, severely hampering the security of the implementation.

Open-source. We have open-sourced VWATTACKER, including the complete VoWiFi network testbed, the LLM-based property extraction technique, and the oracles to foster research in this area and help vendors test their UEs[1].

Responsible disclosure. As our findings can result in real-world attacks, we have responsibly disclosed the results of our work to all the related vendors (i.e., baseband vendors and UE vendors). One issue has been acknowledged as CVE-2025-20667.

Contributions. We summarize our contributions as follows:

- We propose VWATTACKER, the framework for analyzing the security of VoWiFi implementations. VWATTACKER includes a *complete* VoWiFi network testbed for COTS UEs. To the best of our knowledge, VWATTACKER is the *first* systematic security framework to test VoWiFi UE implementations.
- To reduce manual labor and increase the scalability of property-guided testing, we design TESTGEN, an LLM-based semi-automatic property extraction technique based on in-context learning and RAG. Using this approach, VWATTACKER extracts 63 properties from the 11 specifications, which results in 1,116 ATCs after applying two types of transformations.
- We test a total of 21 COTS UEs from 14 vendors. Our deterministic oracles uncover 10 issues. Based on vulnerabilities, we reveal 3 new attacks.

II. VOICE OVER WIFI

This section provides an overview of the VoWiFi protocol.

VoWiFi architecture. VoWiFi is a Voice over IP (VoIP) service run by a mobile network. To use the service, the *User Equipment* (UE) sets up a Virtual Private Network (VPN) with the mobile network after it attaches to a WiFi access point (AP). The VPN peer that communicates with the UE on the mobile network side is called an *evolved Packet Data Gateway* (ePDG). It is a connection point between the Internet and the mobile network. Once the VPN is established, the UE starts communicating with the *IP Multimedia Subsystem* (IMS), the main components of which are the *Proxy Call Session Control Function* (P-CSCF), the *Interrogating Call Session Control Function* (I-CSCF), and the *Service Call Session Control Function* (S-CSCF). The P-CSCF is an endpoint that directly communicates with the UE by exchanging the SIP messages. It forwards a request from the UE to the I-CSCF that is responsible for selecting the S-CSCF to be assigned for the session. The S-CSCF is mainly responsible for registration. We refer to the participants in the VoWiFi architecture (i.e., UE, ePDG, and IMS) as *VoWiFi entities* (or *entities*). These

[1]https://github.com/hw5773/vowifi-ue-testing-framework

entities execute the IKE protocol, the EAP-AKA protocol, and the SIP protocol, respectively.

IKEv2 [17]. It is the subprotocol used to establish the VPN between the UE and an ePDG. The VPN is set up with a common security association, including cryptographic algorithms and keys. IKEv2 proceeds with several *exchanges*; each such exchange consists of a *request* message and a *response* message. In other words, a request message sent by one entity is always followed by a response message from the other entity. In the VoWiFi scenario, the UE always initiates IKEv2 by sending the `IKE_SA_INIT` request message to the ePDG, which responds with the `IKE_SA_INIT` response message. The `IKE_SA_INIT` exchange is responsible for establishing the IKE security association (SA) that contains cryptographic keys and algorithms. Then, the protocol is followed by several `IKE_AUTH` exchanges that are responsible for authenticating each other through EAP-AKA and establishing a child SA.

EAP-AKA [5]. It is the subprotocol used for mutual authentication between the UE and the ePDG, utilizing a challenge-response mechanism and symmetric cryptography. The UE and the ePDG exchange AKA messages encapsulated with the `IKE_AUTH` exchanges. In the VoWiFi protocol, EAP-AKA is initiated after the ePDG receives the first `IKE_AUTH` request message from the UE, which includes its identifier (i.e., International Mobile Subscriber Identity, IMSI) and the intended peer's identifier (i.e., ims). The ePDG fetches the key materials through the UE's IMSI and responds with the `EAP-Request/AKA-Challenge` message that contains a random number (i.e., `AT_RAND`), an authentication token (i.e., `AT_AUTN`), and a message authentication code (i.e., `AT_MAC`). The UE verifies both `AT_AUTN` and `AT_MAC` and sends `AKA-Client-Error` if any of them is invalid. Otherwise, the UE encrypts the random number with its unique key and sends the encrypted message (i.e., `AT_RES`) with a message authentication code (i.e., `AT_MAC`) in the `EAP-Response/AKA-Challenge` message. If the ePDG successfully validates `AT_RES`, which means that the UE is authenticated, the ePDG sends the `AKA-Success` message to the UE, and the IKE channel is finally established between the UE and the ePDG.

SIP [34]. It is the subprotocol used to register a UE with the IMS or to make and receive a VoWiFi call. As we focus on the SIP registration in this paper, we only describe the registration flow below, which is based on the challenge-response protocol. Once the IKE channel is established, the UE sends the `SIP REGISTER` message to the IMS server through the ePDG. IMS responds with the `401 Unauthorized` message that includes a nonce value. The UE generates the authentication token with the nonce and responds with the second `SIP REGISTER` message which contains the token. Finally, the UE is authenticated and registered if the token is valid, and IMS sends a confirmation message (i.e., `200 OK`). Note that the SIP messages are encrypted and integrity-protected with the child SA established during the IKE protocol.

III. Overview of VWAttacker

This section provides an overview of the VWAttacker (see Figure 1) with its threat model, challenges, and requirements.

A. Threat Model

In adversarial testing, we assume a Dolev-Yao attacker [9]. The attacker can eavesdrop, modify, or drop any message and inject new messages. However, the attacker is computationally bounded; that is, the attacker cannot break cryptographic assumptions. Therefore, the attacker cannot decrypt an encrypted message unless they possess the decryption key. This is a realistic threat model to uncover protocol-related issues and is heavily used by different protocol testing frameworks [31].

In the context of VoWiFi, we assume that the attacker resides on the communication path between the VoWiFi UE and the VoWiFi core network. This includes malicious WiFi access points, compromised routers, or untrusted intermediaries that can intercept and manipulate network traffic. For instance, the attacker can impersonate a legitimate WiFi network (e.g., Evil Twin AP) to intercept VoWiFi signaling and media traffic [4], [21], [28], [32]. However, the attacker does not compromise the UE itself and cannot access protected memory or internal protocol state of the VoWiFi implementation.

B. Challenges and Requirements

The critical challenges in the design of VWAttacker are:

(C1) VoWiFi network testbed. As commercial VoWiFi UEs are entirely black-box, designing a security testing framework requires a complete UE testbed to interface with them. Unfortunately, no open-source VoWiFi UE testbed is available for us to use. This is a fundamental challenge that prevents VoWiFi implementation testing. Furthermore, the testbed should be fully customizable to implement a systematic framework for adversarial testing and should allow UEs to execute the complete VoWiFi procedures.

(C2) Scalability of property-guided testing. To comprehensively analyze the VoWiFi implementations and improve the scalability of property-guided testing, we need to tackle two sub-challenges:

(C2.1) Scalable and accurate property extraction: Previous works on property-based adversarial testing rely on manual property extraction [20], [31], which is time-consuming, error-prone, and limited in scope. Although recent work explores LLM-assisted fuzzing [26] for other domains, these approaches typically lack integration with wireless protocol semantics or are optimized for application-layer protocols. Moreover, they do not address the ambiguities or underspecifications common in VoWiFi standards, which can lead to hallucinations. Hence, a specialized approach is needed to extract low-level protocol properties from fragmented specifications (3GPP, RFCs, conformance suites) while preserving contextual correctness.

(C2.2) Property-violation detection: Another crucial part of property-guided testing is detecting when the property has been violated. As the VoWiFi protocol does not include a formal canonical model, it is impossible to directly compare the behavior with a canonical model to detect deviations and property violations. Therefore, we must design side-channel oracles that automatically detect property violations.

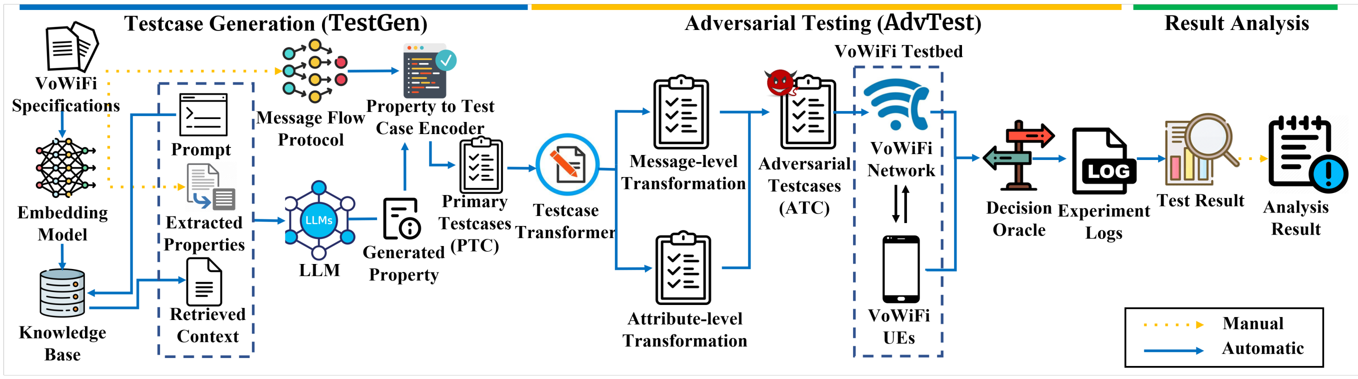

Fig. 1: Overview of VWATTACKER.

C. Addressing the Challenges

To address (**C1**), we build a VoWiFi network testbed that provides commercial UEs with VoWiFi service by integrating existing open-source subprotocol implementations, such as *StrongSwan* and *Kamailio*, together with a custom ISIM card, *sysmoISIM-SJA2*. In deploying a completely VoWiFi-supporting testbed, we aim to support diverse devices without requiring any modification on them (e.g., no rooting) and extra hardware. Due to these requirements, customizing the VoWiFi network needs considerable engineering effort. We detail how we develop the testbed in Section IV.

To address (**C2.1**), we design an LLM-based property extraction and testcase generation framework, called TESTGEN, that extracts diverse properties from the VoWiFi specifications and generates testcases in a semi-automated fashion. LLMs can learn from in-context information and utilize this information to perform specific downstream tasks without requiring retraining or modifying the model weights. With VoWiFi specifications as context and some manually extracted properties as examples, we use an LLM to extract properties from VoWiFi specifications. Although modern LLMs are likely pre-trained on many publicly available protocol specifications, relying solely on their internal memory can lead to hallucinations or overlook domain-specific details scattered across loosely connected sections of the specifications. To tackle this challenge, we utilize Retrieval Augmented Generation (RAG). By explicitly grounding the model with relevant context at query time, RAG ensures that the LLM does not have to rely on memorization from massive pretraining corpora, but instead focuses on interpreting the specification fragments most relevant to the prompt. This makes the extraction process both more targeted and transparent, especially in a domain like VoWiFi where specifications are complex and modular [21]. After extracting the properties, an encoder converts these extracted properties into the primary testcases (PTCs) using the message flows of related protocols. Two transformation techniques are then applied to these PTCs, resulting in adversarial testcases (ATCs), which are used to test the security of UEs in the customized VoWiFi testbed. To address (**C2.2**) and automatically detect property violations, we design two types of decision oracles that retrieve logs generated from each test and raise alerts whenever they find issues. These are used to detect semantic bugs or availability issues.

D. High-Level Overview of VWATTACKER

At a high level, VWATTACKER is divided into three modules (see Figure 1): (i) testcase generation (TESTGEN); (2) adversarial testing (ADVTEST); and (3) result analysis.

TESTGEN generates testcases in two parts:

- **(1) Property extraction using LLMs:** Initially, some properties are manually extracted and provided as examples to guide the LLM in understanding the type of properties to look for in the specifications. Using the provided context and examples, the LLM processes the VoWiFi specifications to identify and extract a comprehensive set of properties.
- **(2) Property-to-testcase encoding:** The properties extracted by the LLM are fed into the property-to-test-case encoder, along with the message flows of the VoWiFi protocol. The encoder translates the properties into PTCs, which are then used to automatically test the VoWiFi system in a dedicated testbed, ensuring comprehensive coverage and efficient validation.

The second step, ADVTEST, has three main parts:

- **(1) Testcase transformer:** We mutate the PTCs using the two transformations, creating ATCs. As messages containing attributes are exchanged in the main subprotocols of VoWiFi, namely IKEv2 and SIP, we design two types of transformations with mutation targets at different levels. The *message-level* transformation inserts, replaces, drops, or replays a target *message*, while the *attribute-level* transformation inserts, updates, or drops an *attribute* in a target message. After applying these transformations to PTCs, we get ATCs.
- **(2) VoWiFi UE testing:** ATCs are sent to the controller of the VoWiFi testbed. The controller processes them, directs VoWiFi entities to send messages according to ATCs, receives the responses and outputs the logs.
- **(3) Decision oracles:** The decision oracles retrieve logs and report "positives" if they find UEs' misbehavior due to adversarial testing or their deadlock states.

In the final step, result analysis, we run scripts to summarize the experiment logs into the test results. Then, we manually inspect the test results and investigate the root causes of UEs' misbehavior or deadlock states.

IV. VoWiFi Network Testbed

A. Requirements

To support a comprehensive black-box property-guided adversarial testing framework for VoWiFi, the underlying testbed should satisfy the following requirements: **(R1) Device-diversity:** The testbed should be able to work with diverse UEs. Satisfying this requirement is challenging, as VoWiFi implementations show quite different behavior. For instance, some UEs always send the `IKE_delete` message before turning off the WiFi interface, while others do not send the message. **(R2) Plug-and-play:** The testbed should not require any change in UEs and should be plug-and-play. For instance, if the testbed were to require rooting the UEs, it would entail manual effort on the part of testers, and the usage of the testbed would be minimal. **(R3) Malleable testing system:** Once we have the VoWiFi network testbed, we can send any test input to VoWiFi UEs. To evaluate the security of VoWiFi UEs in a wide range of scenarios, the testbed should support testing any possible scenario provided by the tester. This, in turn, requires generating a stateless system from a highly stateful protocol. To make the testbed useful, all VoWiFi entities should be directly controlled, and log messages from these entities should be collected in a unified format.

B. High-level Design

To design the testbed, we integrate open-source subprotocol implementations, such as *StrongSwan* and *Kamailio*, together with a custom ISIM card, *sysmoISIM-SJA2*, and make 21 UEs from 14 different vendors runnable on the testbed satisfying (**R1**). Our testbed operates in a plug-and-play fashion, requiring no modifications to the devices or additional hardware, following (**R2**). We provide a simple, JSON-based interface for testers to describe how they want to evaluate a target UE. We define a testcase (TC) as a series of commands by which the tester can make VoWiFi entities send specific messages (e.g., making ePDG send the second `IKE_AUTH` response message with the value of the EAP failure in EAP-AKA) according to the test UE's messages. For creating the malleable testbed following (**R3**), we design it based on a control architecture following the *central-peripheral* paradigm, where a controller directs VoWiFi entities (i.e., UE, ePDG, and IMS) through a UE agent, an ePDG agent, and an IMS agent, respectively. To achieve this, we move all the logic away from the peripheral agents to the Controller to create a stateless system.

C. Control architecture

Our control architecture (see Figure 2) consists of a Controller that processes a TC and Agents (UEAgent, ePDGAgent, and IMSAgent) that communicate with the Controller to control VoWiFi entities (i.e., UE, ePDG, and IMS). Based on the Controller and Agents, VWAttacker takes a TC described by a series of commands as input and outputs a log, which is a series of reports from Agents. To direct VoWiFi entities according to a TC, we use a *command-report* protocol between the Controller and the Agents. In detail, the Controller sends a command to an agent to control the

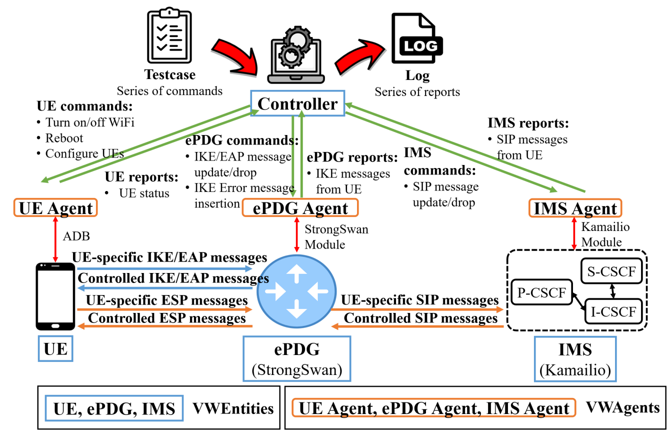

Fig. 2: The control architecture of the Testbed

corresponding VoWiFi entity. Then, the agent reports the result of the command (e.g., UE's status or responding messages from UE) for which the agent is responsible to the Controller.

V. Details of VWAttacker

This section describes how we design and implement VWAttacker leveraging the VoWiFi testbed. Since the result analysis module is trivial, in the following subsections, we present the details of TestGen and AdvTest–which are the novel elements of VWAttacker.

A. TestGen*: Testcase Generation*

TestGen consists of two components: (1) the LLM-based property extractor and (2) the property-to-testcase encoder. The former contributes to extracting properties from specifications, while the latter encodes the properties into PTCs.

LLM-based property extractor. To automatically extract security-relevant properties from loosely structured VoWiFi specifications, we build a Retrieval-Augmented Generation (RAG) framework tailored for this domain. The framework begins by embedding 11 specification documents, including 3GPP technical reports, IETF RFCs and conformance test specifications. These embedded chunks are indexed for efficient semantic retrieval. We then use a retrieval module to fetch specification fragments relevant to a given query. Along with a small set of manually curated example properties (created in approximately 1 hour by a domain expert), these retrieved documents are provided as context to a prompted large language model (LLM). The model then generates candidate properties grounded in the retrieved content. Although LLMs are likely pretrained on publicly available specifications, relying on their internal knowledge can lead to inconsistent outputs, especially for rarely cited clauses or ambiguous conditions. Using RAG avoids these pitfalls by explicitly providing relevant specification fragments at inference time, making the extraction more precise, consistent, and explainable. Compared to fine-tuning or naive prompting, RAG offers a flexible and modular solution. It improves performance (see Section VI) without requiring retraining or labeled datasets, and enables us to trace each generated property back to specific specification fragments.

Property-to-testcase encoder. The encoder takes the generated properties, the RAG-based model used for property extraction and the protocol message flow as input and converts the properties into PTCs. To achieve this, the encoder first identifies the message name from the property and matches it to the protocol's message flow to determine the state required to execute the property. Then, following this state, the encoder creates a series of commands by appending messages with related VoWiFi entities one after another, based on the protocol's message flow. For each message, the relevant fields and values (if not specified in the property) are generated by querying the RAG-based LLM model. Note that, not all the properties include an explicit message name and refers to message fields; in those cases, the message names are manually provided to the encoder.

B. ADVTEST*: Adversarial Testing*

ADVTEST begins with the testcase transformer that converts PTCs into ATCs. Then, ATCs are fed into the VoWiFi testbed for the adversarial testing. Finally, the misbehavior of UEs is automatically detected with the help of the decision oracles.

Testcase transformer. For generating the ATCs, we develop transformations at two different levels: message-level and attribute-level. A *message-level transformation* changes the final command of the PTC to generate an adversarial message. There are two types of message-level transformations: ❶ **Message substitution:** final command is set to send an error message (e.g., an invalid SPI) to see how a UE reacts to unexpected error messages. ❷ **Message replay:** final command is manipulated to send one of the previous messages again to check how a UE behaves against the replay attack. Next, an *attribute-level transformation* is to modify or remove one specific attribute-value pair of the final command. There are two types of attribute-level transformation, which are: ❶ **Attribute value update:** changes the value of an attribute in the PTC. The target attributes which revise values, include the length of a message, the protocol version, algorithms to be used, sequence numbers, and others. We also extract a set of possible values for each attribute from the LLM and generate TCs based on the set for each attribute. For example, we utilize the LLM to make a possible set for the encryption algorithm in the IKE protocol containing values [3–28] and -1 because the former is specified in the specification, and the latter is an exception. ❷ **Attribute drop:** drops a specific attribute from the message. The attributes to be dropped include all the attributes in the message, such as the algorithms to be used or the protocol version. For example, we drop the encryption algorithm field in the IKE_SA_INIT response message.

VoWiFi UE testing. The ATCs generated from the first step are fed into the CONTROLLER to evaluate the security of the implementations. The CONTROLLER takes a set of ATCs and processes them one by one. According to each ATC, the CONTROLLER controls AGENTS to send messages through the command-report protocol. The logs are finally collected on the CONTROLLER.

Decision oracles. To automatically determine UE misbehavior, we introduce two decision oracles that work over the collected logs and analyze whether the UE's behavior is far from what we expect. These oracles are: ❶ **Function oracle:** it raises an alert if a UE responds to an adversarial message. There are three behaviors that a UE can perform when the ATC is executed: (1) ignore the adversarial message, (2) respond with the positive message that the UE usually replies to the non-adversarial message, and (3) respond with the negative message indicating an error. In case of (1), the oracle determines that the UE behaves correctly; thus, it does not raise an alert. When the UE responds with a positive message (the case (2)), the oracle raises an alert as it means that the UE violates the tested property. Finally, when the UE responds with a negative message (the case (3)), the oracle also raises an alert because there is the possibility of incorrect behavior. ❷ **Liveness oracle:** it checks whether a UE can re-execute the VoWiFi protocol after it receives an adversarial message and disconnects the corresponding session. We expect that the adversarial message should not affect the other sessions. Therefore, the oracle raises an alert if the UE does not send the IKE_SA_INIT request message to an ePDG after the aborted session.

With the above two oracles, we design the testing for each testcase as follows: ❶ **Running an ATC:** VWATTACKER performs adversarial testing based on an ATC for a specific UE. ❷ **Checking an anomalous message flow:** The function oracle checks the logs and raises an alert if it sees an unexpected message from the UE. ❸ **Running a normal registration:** VWATTACKER tries to run the normal VoWiFi execution with the UE. ❹ **Checking liveness:** The liveness oracle checks if the UE executes the VoWiFi protocol correctly. To resolve any non-determinism due to over-the-air testing, this process is repeated three times for each ATCs to get stable results. Note that the execution of the normal registration with the liveness oracle is necessary not only to simply check whether the UE is alive but also to understand the result of the function oracle in more detail. The reason is that we assume that a UE should not respond to an adversarial message. However, we cannot determine whether the lack of response by the UE is due to a crash of the UE, which would represent a false negative of the function oracle. Therefore, we use the liveness oracle to catch the case of the crash.

Finally, the logs on each ATC are labeled with the flags from the decision oracles. We manually analyze the results to pinpoint the issues and the root causes.

VI. EVALUATION

To evaluate the performance of VWATTACKER, we aim to answer the following research questions:

- **RQ1.** How faithful and relevant are the properties generated by TESTGEN?
- **RQ2.** How effective is VWATTACKER in finding issues and attacks on diverse VoWiFi UE implementations?
- **RQ3.** How effective is VWATTACKER, compared to the previous IKE or SIP implementation testing approaches, and the conformance test suite from the VoWiFi standards [3]?

Model	Generation Performance														Retrieval Performance		
	AR↑		HR↓		BLEU↑		METEOR↑		ROUGE-L↑						CP↑	CRec↑	CRel↑
									Precision		Recall		F1-score				
	R	NR	R	NR	R	NR	R	NR	R	NR	R	NR	R	NR			
Mistral-7B-v0.1	0.83	0.68	0.30	0.30	0.42	0.18	0.67	**0.43**	**0.77**	**0.53**	0.51	0.37	0.57	0.39	0.62	0.51	0.50
Llama-3-8B-Instruct	0.84	**0.94**	0.05	0.31	0.50	0.22	0.64	0.42	0.65	0.48	0.76	0.49	0.68	**0.46**	0.92	**0.61**	**0.69**
Llama-3.1-8B-Instruct	**0.90**	0.85	**0.04**	**0.05**	**0.53**	**0.24**	**0.68**	**0.43**	0.68	0.45	**0.84**	**0.52**	**0.73**	**0.46**	**0.93**	0.56	0.66

TABLE II: Evaluation results of TESTGEN (CP: Contextual Precision, CRec: Contextual Recall, AR: Answer Relevancy, HR: Hallucination Rate, R: RAG, NR: No RAG, ↑: higher is better, ↓: lower is better)

A. Performance Evaluation of TESTGEN

We evaluate TESTGEN in two dimensions – (1) generation performance and (2) retrieval performance – and report the results (see Table II).

Generation Performance. The generation performance evaluates the quality of properties that a model produces. To evaluate generation performance, we use the metrics listed below: ① **Answer relevancy (AR):** it is used to assess whether the LLM returns concise answers by determining the proportion of sentences in the LLM output that are relevant to the input. ② **Hallucination rate (HR):** it measures the proportion of hallucinated sentences in an LLM output as this is critical when dealing with LLM-generated outputs. ③ **Bilingual evaluation understudy (BLEU) score:** it measures the overlap between the generated text and the reference text (i.e., ground truth). ④ **Recall-oriented understudy for gisting evaluation (ROUGE) score:** it measures the overlap of n-grams between the generated text and the reference text. ⑤ **Metric for evaluation of translation with explicit ordering (METEOR) score:** it provides a more nuanced evaluation than BLEU when considering synonyms and stemming.

Our results show that RAG effectively improves the generation performance. Further, the advanced models demonstrate higher scores in generation. Table II shows that Llama 3.1 performs best in almost all the metrics discussed above, especially in reducing the HR, which is very important for high-quality text generation. We find adding context retrieved from the specification knowledge base further improves the generation performance.

Retrieval Performance. As the property generation largely relies on the retrieved context, we measure the retrieval performance based on the following metrics: ① **Contextual precision (CP):** it measures how accurately the retrieved context aligns with the generated text. This metric is calculated as the ratio of relevant information in the generated text to relevant information in the retrieved context. ② **Contextual recall (CR):** it determines the proportion of sentences in the expected output or ground truth that can be attributed to the retrieved context. ③ **Contextual relevancy (CR):** it is the proportion of sentences in the retrieved context relevant to a given input. To understand generation performance in detail, we measure retrieval performance on the models. We find that the main difference between models in retrieval performance is CP. Thus, our conclusion is that the higher generation performance of the advanced models is mainly due to the higher CPs.

Choice of Models. We work exclusively with open-source models to ensure transparency and flexibility. After performing basic sanity checks across various open-source LLMs, we select Mistral-7B-v0.1, Llama-3-8B-Instruct, and Llama-3.1-8B-Instruct for in-depth analysis due to their strong performance and alignment with our requirements. These models demonstrate reliable outputs during preliminary evaluations, making them suitable candidates for further exploration within the RAG framework. As we leverage the RAG approach, TESTGEN does not require fine-tuning the LLMs.

B. Result of Adversarial Testing

We generate 1,116 ATCs based on 63 security properties extracted from TESTGEN. We find the 10 issues discovered by ADVTEST on 21 UEs (disclosed issues in Table III, and the lists of UEs per issues in Table IV).

Support of Weak Algorithms in IKE (Issues #1 – #5). In the `IKE_SA_INIT` exchange, *the deprecated algorithms MUST NOT or SHOULD NOT be implemented* [29]. We conduct adversarial testing against the property, seeing if a UE establishes the `IKE_SA` with such weak algorithms.

ATC **and VWATTACKER behavior.** Our transformation generates ATCs by setting the operation to update for the attribute security_association under the message `IKE_SA_INIT` response. The values for the algorithms are automatically extracted by TESTGEN referring to the specifications. Finally, the ATC is sent to the attached UE. The function oracle raises alerts when UE responds to the adversarial message, which shows the `IKE_SA` is established with the algorithm.

Result and analysis. We analyze the logs to see if there is any `IKE_SA` established with weak algorithms, which are DES, 3DES (encryption algorithms), AUTH_HMAC_MD5_96 (an integrity algorithm), PRF_HMAC_MD5 (a pseudorandom function), 1536-bit MODP, 1024-bit MODP, 768-bit MODP, 1024-bit MODP with 160-bit prime order, 2048-bit MODP with 224-bit prime order, and 2048-bit MODP with 256-bit prime order. Although 3DES is not officially deprecated in the domain of IKEv2, we set it as a weak algorithm since it is getting deprecated in another domain [15]. We find that 9–15 UEs establish weak `IKE_SA`s. Note that the results show the possibility of a weak channel that can be established with an incorrectly configured ePDG [11]. Also, the results report the UEs that do not comply with the specifications.

Support of Weak Algorithms in SIP (Issues #6 – #7). In the `401_unauthorized` message, the SIP authentication algorithms and the encryption/integrity algorithms are negotiated. We conduct adversarial testing to see if UE establishes an SIP session with weak algorithms.

ATC **and VWATTACKER behavior.** Our transformation generates ATCs by setting the operation to update the attributes under WWW-Authenticate and Security-Server, respectively. The possible values extracted from specifications are assigned to

No.	Testcase	Result
Support of Weak Algorithms in IKE		
1	Modify the value specifying the weak encryption algorithm in the IKE_-SA_INIT response	The IKE_SA is established with a weak encryption algorithm (e.g., DES)
2	Modify the value specifying the weak integrity algorithm in the IKE_SA_-INIT response	The IKE_SA is established with a weak integrity algorithm (e.g., AUTH_-HMAC_MD5_96)
3	Modify the value specifying the weak pseudorandom function in the IKE_-SA_INIT response	The IKE_SA is established with a weak pseudorandom function (e.g., PRF_-HMAC_MD5)
4	Modify the value specifying the weak DH group and key in the IKE_SA_-INIT response	1) The weak DH key (e.g., 1024-bit MODP) is advertised from UE by default and 2) the IKE_SA is established with the key
5	Substitute the INVALID_KE_PAYLOAD with a weak DH group for the IKE_SA_INIT response	The IKE_SA is established with a weak DH group (e.g., 1024-bit MODP with 160-bit prime order)
Support of Weak Algorithms in SIP		
6	Modify the SIP authentication algorithm in the 401_unauthorized message	The weak SIP authentication algorithm is run (e.g., MD5)
7	Modify the encryption algorithm for the SIP messages in the 401_-unauthorized message	The weak pair of the encryption/integrity algorithm is advertised from UE and it is selected (e.g., DES/HMAC_MD5)
Zero DH Key with IKE_SA_INIT		
8	Drop the key exchange payload from the IKE_SA_INIT response	The DH shared secret is set to 0, disclosing keys for an attacker to decrypt the first two IKE_AUTH messages
Nonce Bypass with IKE_SA_INIT		
9	Drop the nonce payload from the IKE_SA_INIT response	The nonce value from the ePDG is set to 0 in a UE
DH Group Downgrade		
10	Substitute the INVALID_KE_PAYLOAD with a *weaker* DH group than a UE-advertised group for the IKE_SA_INIT response	The DH group is downgraded from the initially advertised one to the weaker one (e.g., 2048-bit MODP $\rightarrow$ 1024-bit MODP)

TABLE III: List of issues disclosed by VWATTACKER

these attributes. VWATTACKER runs the ATCs and the function oracle raise an alert if the algorithm is selected.

Result and analysis. We analyze the logs to see if any weak algorithm is selected. The algorithms of our interest include MD5 (SIP authentication), a pair of DES (encryption algorithms), and HMAC_MD5_96 (an integrity algorithm). Although these algorithms are not explicitly deprecated, they are controversial [18], [27] in using them due to their weaknesses. There are 2 UEs responding with MD5 when they receive the 401_unauthorized message with MD5. Among them, we find only one UE supports MD5, while the other simply copies the name of the algorithm from 401_unauthorized and sends the response message that sets the algorithm name to be the copied name. Interestingly, we find that all the UEs advertise the use of DES as an encryption algorithm together with HMAC_MD5_96 as an integrity algorithm. Note that this result demonstrates the impact of a unified testbed. Without completing the IKEv2 protocol, the implementation does not reach this state to trigger this SIP issue.

Zero DH Key with IKE_SA_INIT (Issue #8). In the initial exchange (i.e., IKE_SA_INIT), The IKE_SA_INIT response message must contain the key exchange payload that contains the responder's DH key [17]. We check how a UE responds to the message against the property.

ATC **and VWATTACKER behavior.** An ATC is generated by setting the operation to drop for the attribute key_exchange under the message IKE_SA_INIT response. When VWATTACKER receives the ATC, the EPDGAGENT removes the key exchange payload from the generated IKE_SA_INIT response message and sends it to the attached UE.

Result and analysis. The function oracle raises alerts in the experiment logs of 8 UEs as the UEs respond to this revised message with the encrypted first IKE_AUTH request message, which should not happen. We verify that we can decrypt the message with the 0s of the DH shared secret.

Threat model and attack. The first IKE_AUTH message contains the UE's IMSI which can be used to track a specific UE, an attacker can know the value by decrypting the message on-the-fly. To perform this attack, an attacker needs to know the target's SPI and forge the IKE_SA_INIT response message without the key exchange payload. The threat model is practical as IKE_SA_INIT has no integrity protection. When an attacker successfully sends the IKE_SA_INIT message without the key exchange payload, a UE sets a DH shared secret to 0 and extracts encryption and integrity keys for the IKE channel. Then, a UE sends the first IKE_AUTH message encrypted with the keys. An attacker can know a UE's IMSI by decrypting the message. Although previous work reports that the VoWiFi protocol is vulnerable to IMSI catching attack [6], our finding is a more implementation-specific issue and makes the IMSI catching attack easier without leveraging the DNS infrastructure. Through this attack, in the affected implementations, it is possible for an attacker to cause a DH key security bypass and steal devices' IMSI.

Nonce Bypass with IKE_SA_INIT (Issue #9). In the initial exchange (i.e., IKE_SA_INIT), *the IKE_SA_INIT response message must contain the nonce payload* [17] to establish the DH shared key and to avoid the replay attack.

ATC **and VWATTACKER behavior.** Our transformation generates an ATC by setting the operation to drop for the attribute nonce under ike_sa_init. The VWATTACKER directs the EPDGAGENT to remove the nonce payload from the IKE_SA_INIT response message and to send it to the UE.

Result and analysis. We find that the 8 UEs respond to this revised message, caught by our function oracle. The UE accepts IKE_SA_INIT and sets the nonce value to zero. Therefore, if a man-in-the-middle attacker can send the message with the nonce payload earlier than the sender, the attacker can successfully set the nonce to zero, substantially reducing the entropy of the secure channel, which is undesirable.

DH Group Downgrade (Issue #10). According to [17], the INVALID_KE_PAYLOAD error message that contains a specific DH group triggers that *the initiator MUST retry the IKE_SA_INIT*

Issue	# of UEs	Affected UEs
1	9 (DES)	HTC U11 life, LG Stylo 6, Moto e5 plus, Nokia G100, OnePlus 9R, OnePlus Nord 20, Pixel 4a, UMIDIGI A13 Pro, ZTE Stage 5G
	15 (3DES)	BlackCyber I14, BlackCyber I15, Blackview A55, HTC U11 life, LG Stylo 6, Moto e5 plus, Nokia G100, NUU B15, OnePlus 9R, OnePlus Nord 20, Pixel 4a, TCL 40XL, Ulefone Note 14, UMIDIGI A13 Pro, ZTE Stage 5G
2	15 (HMAC_MD5_96)	BlackCyber I14, BlackCyber I15, Blackview A55, HTC U11 life, LG Stylo 6, Moto e5 plus, Nokia G100, NUU B15, OnePlus 9R, OnePlus Nord 20, Pixel 4a, TCL 40XL, Ulefone Note 14, UMIDIGI A13 Pro, ZTE Stage 5G
3	6 (PRF_HMAC_MD5)	BlackCyber I14, BlackCyber I15, Blackview A55, NUU B15, TCL 40XL, Ulefone Note 14
4	12 (1024-bit MODP)	BlackCyber I14, BlackCyber I15, Blackview A55, NUU B15, TCL 40XL, Ulefone Note 14
5	19 (1024-bit MODP)	HTC U11 life, LG Stylo 6, Moto e5 plus, Nokia G100, OnePlus 9R, OnePlus Nord 20, Pixel 4a, UMIDIGI A13 Pro, ZTE Stage 5G
	15 (1536-bit MODP)	BlackCyber I14, BlackCyber I15, Blackview A55, HTC U11 life, LG Stylo 6, Moto e5 plus, Nokia G100, NUU B15, OnePlus 9R, OnePlus Nord 20, Pixel 4a, TCL 40XL, Ulefone Note 14, UMIDIGI A13 Pro, ZTE Stage 5G
	6 (1024-bit MODP 160-bit prime order)	BlackCyber I14, BlackCyber I15, Blackview A55, NUU B15, TCL 40XL, Ulefone Note 14
	6 (2048-bit MODP 224-bit prime order)	BlackCyber I14, BlackCyber I15, Blackview A55, NUU B15, TCL 40XL, Ulefone Note 14
	6 (2048-bit MODP 256-bit prime order)	BlackCyber I14, BlackCyber I15, Blackview A55, NUU B15, TCL 40XL, Ulefone Note 14
6	2 (MD5)	LG Stylo 6, Pixel 6a
7	21 (DES/HMAC_MD5_96)	BlackCyber I14, BlackCyber I15, Blackview A55, HTC U11 life, LG Stylo 6, Moto e5 plus, Nokia G100, NUU B15, OnePlus 9R, OnePlus Nord 20, Pixel 4a, Pixel 6a, Samsung Galaxy A21s, Samsung Galaxy A34 5G, Samsung Galaxy A35 5G, Samsung Galaxy S6, TCL 40XL, Ulefone Note 14, UMIDIGI A13 Pro, ZTE Stage 5G
8	8 (No key exchange payload)	BlackCyber I14, HTC U11 life, Moto e5 plus, Nokia G100, OnePlus 9R, OnePlus Nord 20, Pixel 4a, ZTE Stage 5G
9	8 (No nonce payload)	BlackCyber I14, HTC U11 life, Moto e5 plus, Nokia G100, OnePlus 9R, OnePlus Nord 20, Pixel 4a, ZTE Stage 5G
10	1 (2048-bit MODP → 768-bit MODP)	LG Stylo 6
	7 (2048-bit MODP → 1024-bit MODP)	HTC U11 life, LG Stylo 6, Mote e5 plus, Nokia G100, OnePlus 9R, OnePlus Nord 20
	7 (2048-bit MODP → 1536-bit MODP)	LG Stylo 6
	1 (2048-bit MODP → 1024-bit MODP 160-bit prime order)	LG Stylo 6
	1 (2048-bit MODP → 2048-bit MODP 224-bit prime order)	LG Stylo 6
	1 (2048-bit MODP → 2048-bit MODP 256-bit prime order)	LG Stylo 6

TABLE IV: Mapping between the issues and the UEs

Issue (ATC no.)	VWATTACKER	Conformance test suite [3]	Cui *et al.* [8]	SECFUZZ [38]	NSFUZZ [33]	c07-sip [40]
1	●	-	-	-	-	-
2	●	-	-	-	-	-
3	●	-	-	-	-	-
4	●	-	-	-	-	-
5	●	-	-	-	-	-
6	●	-	-	-	-	-
7	●	-	-	-	-	-
8	●	-	-	-	-	-
9	●	-	-	-	-	-
10	●	-	-	-	-	-

TABLE V: Comparison VWATTACKER with other approaches

with the corrected DH group. We test a UE if it resends a key on a weaker, thus downgraded, DH group.

ATC **and VWATTACKER behavior.** Our transformation generates an ATC by setting the operation to substitute for the message `IKE_SA_INIT` response and update for the attribute dh_group of the message. The VWATTACKER directs the EPDGAGENT to send the `INVALID_KE_PAYLOAD` error response message with a downgraded DH group to the UE.

Result and analysis. We find that the 1 – 7 UEs respond to this error message, caught by our function oracle. We see the logs where one UE responds with an arbitrary DH group even if it does not advertise the group. For instance, we can let the UE to send the DH key over the 768-bit MODP group.

Threat model and attack. Based on this issue, an attacker can let a UE to establish a weaker IKE channel. To perform this attack, an attacker needs to be the man-in-the-middle and sends the adversarial `INVALID_KE_PAYLOAD` specified with the weaker DH group to a UE. Concurrently, this attack is also reported in [11] but on different devices. After a weaker IKE channel is established, an attacker can decrypt the messages on the downgraded channel using known attacks [11], [22]

C. Comparison with Previous Work

We compare the capability of VWATTACKER in terms of 1) the ATC generation and 2) the decision oracles in identifying issues with those of other approaches [3], [8], [33], [38], [40] for IKE, SIP, or VoWiFi implementations (see Table I). As the conformance test suite [3] only checks UE's operational behavior, it cannot detect any issues. Cui *et al.* [8] is unable to detect any of the issues detected by VWATTACKER. This is due to several reasons: first, since [8] only focuses on the IKE protocol, it cannot generate any ATCs related to SIP. Second, it cannot create ATCs #5 and #8 because it cannot replace one message with another. This shows why the message-level transformation is useful. Because SECFUZZ [38] can only automatically detect memory corruption in determining issues, it cannot identify any of issues among our findings. NSFUZZ [33] may generate ATCs #6 and #7 as it evaluates *Kamailio* by sending messages with random values for specific attributes, but is not able to detect due to the lack of automated detection using the oracles. Note that C07-SIP [40] cannot identify any issues because it only tests with specific attributes that are not related to our findings.

VII. CONCLUSION AND FUTURE WORK

In this paper, we design a systematic and automated framework called VWATTACKER to analyze the security of VoWiFi UEs. With VWATTACKER, we uncover 10 issues on VoWiFi UE and show our better capabilities over other approaches in generating ATCs and determining issues. As our future work, we aim to include testing for VoWiFi core-network components to cover the entire ecosystem.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their suggestions to improve our paper. This research was supported by NSF under grant 2112471, the KENTECH Research Grant (202200048A), the University of Texas System Rising STARs Award (No. 40071109), and the startup funding from the University of Texas at Dallas.

References

[1] "Voice over wifi (vowifi) market size & share analysis - growth trends & forecasts (2025 - 2030)," https://www.mordorintelligence.com/industry-reports/voice-over-wifi-vowifi-market, 2019, (Accessed on 05/27/2025).

[2] F. 3, "Why is my phone signal so weak indoors? 5 causes & fixes," https://www.hiboost.com/blogs/news/why-is-my-phone-signal-so-weak-indoors-5-causes-fixes-2025-guide, 2025, (Accessed on 1/7/2026).

[3] G. T. 36.523-1, "Evolved universal terrestrial radio access (e-utra) and evolved packet core (epc); user equipment (ue) conformance specification; part 1: Protocol conformance specification release 16," https://www.etsi.org/deliver/etsi_ts/136500_136599/13652301/16.08.00_60/ts_13652301v160800p.pdf, 2021, (Accessed on 09/04/2024).

[4] M. Agarwal, S. Biswas, and S. Nandi, "An efficient scheme to detect evil twin rogue access point attack in 802.11 wi-fi networks," *International Journal of Wireless Information Networks*, vol. 25, no. 2, pp. 130–145, 2018.

[5] J. Arkko and H. Haverinen, "Extensible authentication protocol method for 3rd generation authentication and key agreement (eap-aka)," https://datatracker.ietf.org/doc/html/rfc4187, 2006, (Accessed on 09/05/2024).

[6] J. Baek, S. Kyung, H. Cho, Z. Zhao, Y. Shoshitaishvili, A. Doupé, and G.-J. Ahn, "Wi not calling: Practical privacy and availability attacks in wi-fi calling," in *Proceedings of the 34th Annual Computer Security Applications Conference*, 2018, pp. 278–288.

[7] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2017-2022," https://s3.amazonaws.com/media.mediapost.com/uploads/CiscoForecast.pdf, 2019, (Accessed on 09/06/2024).

[8] Y. Cui, T. Yu, and J. Hu, "Ikev2 protocol fuzzing test on simulated asa," in *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*. IEEE, 2018, pp. 111–116.

[9] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[10] M. E. Garbelini, Z. Shang, S. Chattopadhyay, S. Sun, and E. Kurniawan, "Towards automated fuzzing of 4g/5g protocol implementations over the air," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 86–92.

[11] G. K. Gegenhuber, F. Holzbauer, P. É. Frenzel, E. Weippl, and A. Dabrowski, "Diffie-hellman picture show: Key exchange stories from commercial {VoWiFi} deployments," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 451–468.

[12] GSMA, "The mobile economy," https://www.gsma.com/mobileeconomy/wp-content/uploads/2023/03/270223-The-Mobile-Economy-2024.pdf, GSMA, Tech. Rep., 2024.

[13] S. Herle, "Fhoss github site," https://github.com/herlesupreeth/FHoSS, 2022, (Accessed on 09/05/2024).

[14] S. R. Hussain, I. Karim, A. A. Ishtiaq, O. Chowdhury, and E. Bertino, "Noncompliance as deviant behavior: An automated black-box noncompliance checker for 4g lte cellular devices," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 1082–1099.

[15] B. Kaduk and M. Short, "Rfc 8429: Deprecate triple-des (3des) and rc4 in kerberos," https://datatracker.ietf.org/doc/html/rfc8429, 2018, (Accessed on 09/06/2024).

[16] Kamailio, "Welcome to kamailio - the open source sip server," https://www.kamailio.org/w/, 2001, (Accessed on 09/05/2024).

[17] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen, "Internet key exchange protocol version 2 (ikev2)," https://datatracker.ietf.org/doc/html/rfc5996, 2010, (Accessed on 09/05/2024).

[18] S. Kelly, "Security implications of using the data encryption standard (des)," https://www.rfc-editor.org/rfc/rfc4772.html, 2006, (Accessed on 09/06/2024).

[19] H. Kim, D. Kim, M. Kwon, H. Han, Y. Jang, D. Han, T. Kim, and Y. Kim, "Breaking and fixing volte: Exploiting hidden data channels and mis-implementations," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 328–339.

[20] H. Kim, J. Lee, E. Lee, and Y. Kim, "Touching the untouchables: Dynamic security analysis of the lte control plane," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1153–1168.

[21] H. Lee, I. Karim, N. Li, and E. Bertino, "Vwanalyzer: A systematic security analysis framework for the voice over wifi protocol," in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, 2022, pp. 182–195.

[22] J. Lee, H. Lee, J. Jeong, D. Kim, and T. T. Kwon, "Analyzing spatial differences in the tls security of delegated web services," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 2021, pp. 475–487.

[23] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[24] C.-Y. Li *et al.*, "Insecurity of voice solution volte in lte mobile networks," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 316–327.

[25] Y.-H. Lu *et al.*, "Ghost calls from operational 4g call systems: Ims vulnerability, call dos attack, and countermeasure," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–14.

[26] X. Ma, L. Luo, and Q. Zeng, "From one thousand pages of specification to unveiling hidden bugs: Large language model assisted fuzzing of matter {IoT} devices," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 4783–4800.

[27] G. T.-S. Meeting, "Change request (s3-214307 -r2 1)," https://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_105e/Inbox/Drafts/draft_S3-214307-r2_Securityupdatesforalgorithmsandprotocolsin33.203.docx, 2021, (Accessed on 09/06/2024).

[28] D. Mónica and C. Ribeiro, "Wifihop-mitigating the evil twin attack through multi-hop detection," in *European Symposium on Research in Computer Security*. Springer, 2011, pp. 21–39.

[29] Y. Nir, T. Kivinen, P. Wouters, and D. Migault, "Algorithm implementation requirements and usage guidance for the internet key exchange protocol version 2 (ikev2)," https://datatracker.ietf.org/doc/html/rfc8247, 2017, (Accessed on 09/06/2024).

[30] Osmocomm, "osmo-epdg - vowifi evolved packet data gateway," https://osmocom.org/projects/osmo-epdg, 2024, (Accessed on 09/06/2024).

[31] C. Park, S. Bae, B. Oh, J. Lee, E. Lee, I. Yun, and Y. Kim, "Doltest: In-depth downlink negative testing framework for lte devices," in *USENIX Security Symposium*, 2022.

[32] C. Peng, C.-Y. Li, H. Wang, G.-H. Tu, and S. Lu, "Real threats to your data bills: Security loopholes and defenses in mobile data charging," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 727–738.

[33] S. Qin, F. Hu, B. Zhao, T. Yin, and Zhang, "Registered report: nsfuzz: towards efficient and state-aware network service fuzzing," in *International Fuzzing Workshop (FUZZING)*, 2022.

[34] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "Sip: Session initiation protocol," https://datatracker.ietf.org/doc/html/rfc3261, 2002, (Accessed on 09/05/2024).

[35] R. Sabourin and S. Pemmaraju, "Why indoor coverage is failing businesses—and how to fix it," https://www.ericsson.com/en/blog/2026/1/why-indoor-coverage-is-failing-businesses-and-how-to-fix-it, 2026, (Accessed on 1/7/2026).

[36] J. Shi, S. Wang, M.-Y. Chen, G.-H. Tu, T. Xie, M.-H. Chen, Y. Hu, C.-Y. Li, and C. Peng, "Ims is not that secure on your 5g/4g phones," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 513–527.

[37] strongSwan, "strongswan: the opensource ipsec-based vpn solution," https://www.strongswan.org/, 2005, (Accessed on 09/05/2024).

[38] P. Tsankov, M. T. Dashti, and D. Basin, "Secfuzz: Fuzz-testing security protocols," in *2012 7th International Workshop on Automation of Software Test (AST)*. IEEE, 2012, pp. 1–7.

[39] K. Tu, A. A. Ishtiaq, S. M. M. Rashid, Y. Dong, W. Wang, T. Wu, and S. R. Hussain, "Logic gone astray: A security analysis framework for the control plane protocols of 5g basebands," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 3063–3080. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/tu

[40] C. Wieser, M. Laakso, and H. G. Schulzrinne, "Security testing of sip implementations," 2003.

[41] T. Xie *et al.*, "The dark side of operational wi-fi calling services," in *2018 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2018, pp. 1–1.