2013-11

Correlating GSM and 802.11 Hardware Identifiers

Martin, Jeremy

þÿ C o r r e l a t i n g G S M a n d 8 0 2 . 1 1 H a r d w a r e I d e n t i f i e r s (J e r e m y M a r t i n , D a n n y
þÿ B e v e r l y , a n d J o h n M c E a c h e n (P r o c e e d i n g s o f t h e M i l i t a r y C o m m u n i c a t i o n
þÿ (M I L C O M 2 0 1 3) , (S a n D i e g o , C A , N o v e m b e r 2 0 1 3 .

http://hdl.handle.net/10945/41596

Correlating GSM and 802.11 Hardware Identifiers

Jeremy Martin, Danny Rhame, Robert Beverly, and John McEachen
Naval Postgraduate School
{jbmartin, dsrhame, rbeverly, mceachen}@nps.edu

***Abstract*—The hardware identifiers of common wireless protocols can be exploited by adversaries for both tracking and physical device association. Rather than examining hardware identifiers in isolation, we observe that many modern devices are equipped with multiple wireless interfaces of different physical types, e.g. GSM and 802.11, suggesting that there exists utility in *cross-protocol hardware identifier correlation*. This research empirically examines the feasibility of such cross-protocol association, concentrating on correlating a GSM hardware identifier to that of the 802.11 hardware identifier on the same device. Our dataset includes 18 distinct mobile devices, with identifiers collected over time at disparate locations. We develop correlation techniques from the perspective of two adversaries: i) limited, able to observe identifiers only in time and space; and ii) a more advanced adversary with visibility into the data stream of each protocol. We first test correlation via temporal and spatial analysis using only basic signal collection, mimicking an RF collection with no decryption or data processing capability. Using a constrained optimization algorithm over temporal and spatial data to perform matching, we demonstrate increasing association accuracy over time, up to ≈80% in our experiments. Our second approach simulates the added capability to collect, decrypt, and reconstruct specific application protocol data, and parses the data of one protocol using search terms derived from the other. With the combined techniques, we achieve 100% accuracy and precision.**

I. Introduction

The hardware identifiers of common wireless protocols, e.g. an 802.11 MAC or GSM IMEI, are globally unique and do not change over the lifetime of a device, thereby permitting both tracking and physical device association[1]. As such, these identifiers can be exploited by adversaries for a range of attacks ranging from mobile privacy to targeted denial-of-service. For example, analytic engines frequently use available hardware identifiers [1] for targeted advertising and statistics gathering. More onerously, hardware addresses reveal crucial details about the device that malicious adversaries can leverage. [2] illustrates a GSM air interface attack where the attacker must first know the IMEI of the intended victim, while [3] and [4] demonstrate per-device remote SMS denial-of-service and remote SIM card rooting, respectively. An attacker correlating hardware identifiers, can use details gleaned from protocol *A* to identify and exploit security vulnerabilities inherent to protocol *B*, increasing the available attack vectors.

Rather than examining hardware identifiers in isolation, we observe that many commodity mobile devices, e.g. phones and tablets, are equipped with multiple physically distinct wireless interfaces. This work focuses on an adversary's ability to *correlate GSM and 802.11 hardware identifiers*.

The format and structure of GSM and 802.11 addresses are different and do not facilitate trivial association (e.g. GSM IMEI `490154203237518` and 802.11 MAC `04:0C:CE:C1:AB:4F`). Furthermore, there is no relation between governing identifier allocation authorities. 802.11's unique hardware identifier is an EUI-48 media access control (MAC) address of six bytes [5], as shown in Figure 1, where the most significant three bytes correspond to an Organizationally Unique Identifier (OUI) of the 802.11 chip manufacturer. The remaining three bytes are assigned to be globally unique.

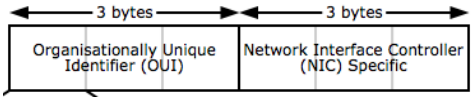

Fig. 1: IEEE 802.11 MAC Address Structure from [5]

GSM utilizes a 15 digit unique identifier known as the International Mobile Equipment Identity (IMEI). As shown in Figure 2, the first six to eight digits represent the Type Allocation Code (TAC), delineating the type and model of the device, while the remaining digits are assigned by the manufacturer to be unique and include a final check digit [6].

TAC	FAC	Serial No	Check Digit
NNXXXX	YY	ZZZZZZ	A

Fig. 2: GSM IMEI Address Structure from [6]

To the best of our knowledge, our work is the first to examine GSM and 802.11 hardware identifier correlation. While Garfinkel et al. identified the 'constellation threat' of tracking an individual's RF device emissions [7], our temporal analysis demonstrates the feasibility of using constellations for cross-correlation. Related work in security and privacy issues surrounding leaked hardware identifiers, location data, and other personal information, e.g. [1], [3], [8], [9], [10], [11], further motivates our work. We build on prior research by demonstrating the power of an adversary to accurately correlate otherwise unrelated hardware identifiers.

We empirically demonstrate GSM and 802.11 hardware identifier correlation using a real-world dataset that includes data captured from 18 distinct mobile devices with identifiers collected over time at disparate locations. We develop correlation techniques from the perspective of two adversaries: i) limited, able to observe identifiers only in time and space; and ii) a more advanced adversary with visibility into the data stream of each protocol. We first test correlation via temporal and spatial analysis using only basic signal collection, mimicking an RF collection with no decryption or data processing capability. Using a constrained optimization algorithm over temporal and spatial data to perform matching, we demonstrate increasing association accuracy over time, up to ≈80% in our experiments. Our second approach simulates the added capability to collect, decrypt, and reconstruct specific application protocol data, and parses the data of one protocol using search terms derived from the other. With the combined techniques, we achieve 100% accuracy and precision.

[1] We leave the analysis of identifier 'spoofing' or obfuscation to future work.

While we focus on GSM and 802.11, our framework may be applied to other protocols (e.g. Bluetooth, CDMA, WiMax), and easily integrates additional data sources and analytic techniques. In this paper, we discuss the IMEI in relation to the original 2G GSM standard. However, the IMEI is used in 3G and 4G GSM-based devices [12], [13], [14]. Thus, the use, allocation, and relevance of the IMEI, and our technique, applies to all of these systems. We hope this work serves as a step forward in identifying a previously under-appreciated privacy and security threat.

II. Data Collection

To evaluate the cross-protocol association techniques we develop, we generate two datasets that reflect data available to the limited adversary (hardware identifiers), and then a richer dataset to model the identifying information available to an advanced adversary (capable of viewing protocol payloads).

A. Limited Adversary Data

We created our dataset using 18 different mobile devices with GSM and 802.11 capability; the devices are shown in Table I, along with a unique identifier which we use to refer to individual devices in this paper (e.g. iPh5 is our 5th iPhone). To model temporal movement, the dataset includes six different snapshots in time, while three different simulated locations model spatial movement (classroom1, classroom2, library). A randomly selected subset of our devices was used for each of the six iterations. We simulate location by randomly assigning each collection test to one of the three locations.

TABLE I: Physical devices used to generate dataset.

Count	Make	Model	ID
2	Acer	Iconia A501	aIa
7	Apple	iPhone 3GS	iPh
1	Apple	iPad	iPa
1	HTC	Hero	hH
1	HTC	Nexus One	hNo
1	HTC	Surround T8788	hSt
2	HTC	Eng Handset	hEh
1	Samsung	I7500	sGa
2	Samsung	19250 Galaxy	sGn

Simulating GSM and 802.11 hardware identifier collection avoids pragmatic issues involved in operating rogue GSM base stations. However, we note that collecting 802.11 identifiers (MAC addresses) is trivial, while obtaining GSM IMEIs is feasible in practice; §IV discusses one possible approach.

The limited adversary has access to the time and location that each IMEI and MAC address is observed for each device. In contrast, the advanced adversary dataset, discussed next, utilizes captured features of real traffic from the 18 devices.

Several public databases contain mappings of various identifiers to device hardware. For example, the first three bytes of an 802.11 MAC address can be queried to determine the device manufacturer. We derived a mapping of OUI to manufacturer from the IEEE database [15].

Similarly, the IMEI can be used to infer the manufacturer and model of the device. We create a mapping of TAC to manufacturer and model correlations derived from various Internet listings and verified via the online TAC lookup service, http://www.nobbi.com/tacquery.php. From the manufacturer and model, we query http://www.gsmarena.com/quicksearch to obtain a list of device capabilities, e.g. 802.11 capability and operating system.

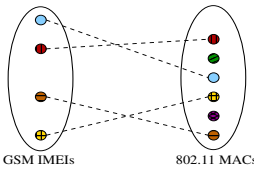

Fig. 3: The correlation problem: associate observed GSM IMEIs with observed 802.11 MACs

B. Advanced Adversary Data

In addition to hardware identifiers observed in space and time, the data available to an advanced adversary includes the full payload of the communication packet stream. Thus, the advanced adversary must address issues of RF propagation and wireless collection considerations, and be able to circumvent any encryption. These concerns are outside the scope of the present research and instead are the focus of other work in high gain antennas, specialized amplifiers, and decryption.

Our advanced adversary dataset therefore effectively eliminates all of these concerns by creating a collection environment where we obtain 100% packet capture. Each device associates with an 802.11 wireless access point under our control where we perform full packet capture.

Within the data payloads of 802.11 frames is a wealth of data. We identify the following widely deployed and readily available protocols and data fields that contain information relevant to the mobile device hardware, including manufacturer and/or device model information:

- User-Agent string in HTTP traffic, which can be used to derive the manufacturer, model, and specific device capabilities and properties.
- User Agent Profiles (UAProf) in the HTTP traffic of some devices, which can reveal the manufacturer, model, and device capabilities.
- Multicast DNS (mDNS) for device discovery, used in Apple's Bonjour, reveals the device-specific hostname.
- Bootstrap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP), which broadcast hostname information indicative of the physical device.

We query a local instantiation of the WURFL database [16] for each collected User-Agent. WURFL maps User-Agents to a profile including the device manufacturer, model, and capabilities. Table II provides example data available to the advanced adversary from four of our devices.

III. Correlation

Abstractly, the correlation problem is bipartite matching whereby we wish to associate observed 802.11 MACs with observed GSM IMEIs based on available evidence (Figure 3). We generalize this correlation as an Integer Linear Program (ILP) that accommodates the different evidence in our datasets as *constraints* on the solution. For example, if TAC i and MAC j were never observed together in the same location, we may reasonably infer that they are an unlikely pair. Conversely, if i and j are observed to be active within the same time window repeatedly, that is an indication that they correspond to the same device.

Let $\mathbf{T}^t$ be the m-by-n matrix representing the t'th snapshot of our temporal data where $|m|$ is the number of distinct GSM TACs and $|n|$ is the number of distinct 802.11 MACs. Thus, $T^t_{i,j}$ is the number of times TAC i and MAC j were

TABLE II: Example data – identifiers collected from 4 of 18 devices in our experiment.

Device	TAC-Derived Info	OUI-Derived Info	BOOTP	Bonjour	UAProf
Acer Iconia A501	Ericsson F5521gw PCIE	Azurewave Tech	n/a	n/a	http://support.acer.com/UAprofile/Acer_A501_Profile.xml
Apple iPhone 3GS	Apple iPhone 3GS 16GB	Apple, Inc	iPhone3GS-1	iPhone3GS-1.local	n/a
HTC Hero	HTC Hero	HTC Corporation	n/a	n/a	http://www.htcmms.com.tw/Android/Common/Hero/ua-profile.xml
Samsung Galaxy Nexus	Samsung I9250 Galaxy Nexus	Samsung Electro	android-cd5db081844aeb9c	n/a	n/a

both active in the previous t time windows. Similarly, let $\mathbf{S}^t$ represent spatial snapshots such that $S^t_{i,j}$ is the number of distinct locations TAC i and MAC j were observed together within the previous t time windows.

Let $\mathbf{A}$ be the sparse association matrix such that $A_{i,j} = 1$ indicates that TAC i is associated with MAC j. We wish to maximize the sum of the "strong" correlations, subject to the feasibility constraints that only one TAC may be associated with at most one MAC and vice-versa. The $\mathbf{A}$ that maximizes the sum of the evidence provides the inferred hardware correlations. As an ILP, which we express in the MathProg modeling language and solve using GLPK [17]:

$$\text{Maximize} \sum_{i=1}^{m}\sum_{j=1}^{n} T_{ij}A_{ij} + S_{ij}A_{ij}$$

$$\text{Subject to} \sum_{i=1}^{m} A_{ij} \leq 1,\ \sum_{j=1}^{n} A_{ij} \leq 1$$

$|m|$ may not equal $|n|$ if there is more of one identifier than the other. Therefore, we constrain the sum of each row vector to be less than or equal to one, i.e. a MAC may or may not be associated with a TAC. Likewise, we constrain the sum of each column vector to be less than or equal to one.

A. Limited Adversary Scenario

We first consider the limited adversary able to detect RF broadcast GSM and 802.11 identifiers.

1) Temporal: Temporal analysis is performed by assigning a value to the number of times an IMEI and a MAC address were seen in collection within the same time window. The hypothesis is that temporal pairings will give significant insight into identifier association without regard to protocol analysis. Possible detractors to this method are limited collection samples, co-located devices, and poor timestamp synchronization between collection platforms. An additional consideration for temporal analysis is the precision of the time window for collection. For our analysis, the time window was known a priori. Future analysis should be conducted to ensure the optimal time window, given the collection capabilities.

2) Spatial: Spatial analysis, although similar to temporal, is different in that we measure the number of disparate locations that the IMEI and MAC address pair were seen together. The hypothesis is that a pair that appears together in different locations has a much higher probability of corresponding to the same device. However, of concern are co-located devices that change locations in similar patterns: for example, three students who move between classrooms in unison throughout the day will have devices with high spatial relationships. We leverage the partial insight from spatial and temporal analysis.

3) TAC – OUI: OUI-based correlation permits only coarse inference as just the device manufacturer is returned from an OUI lookup. In the absence of other protocols for a given pairing, the comparison of the TAC manufacturer and OUI manufacturer should eliminate pairings that cannot be possible.

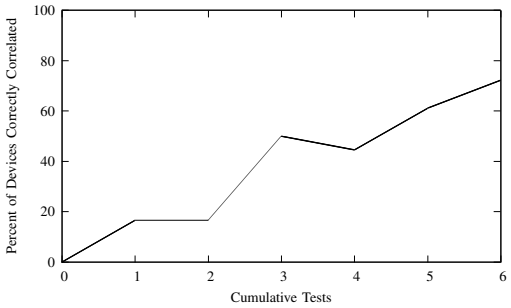

Fig. 4: Devices correctly correlated as a function of time

TABLE III: Results of Temporal, Spatial, Temporal-Spatial (T/S), and Weighted Temporal-Spatial (T*5/S)

Temporal	Spatial	T / S	T * 5 / S
iPh1 = iPh1	iPh1 = iPh1	iPh1 = iPh1	iPh1 = iPh1
iPh2 = iPh2	iPh2 = hEh1	iPh2 = iPh2	iPh2 = iPh2
iPh3 = iPh3	iPh3 = iPh3	iPh3 = iPh3	iPh3 = iPh3
iPh4 = iPh5	iPh4 = hEh2	iPh4 = iPh4	iPh4 = iPh5
iPh5 = iPh4	iPh5 = iPh5	iPh5 = iPh5	iPh5 = iPh4
iPh6 = iPh6	iPh6 = iPh6	iPh6 = iPh6	iPh6 = iPh6
iPh7 = iPh7	iPh7 = hNo1	iPh7 = iPh7	iPh7 = iPh7
iPa1 = iPa1	iPa1 = iPh2	iPa1 = iPa1	iPa1 = iPa1
sGn1 = sGn1	sGn1 = iPa1	sGn1 = sGn1	sGn1 = sGn1
sGn2 = sGn2	sGn2 = hH1	sGn2 = hNo1	sGn2 = sGn2
hSt1 = hSt1	hSt1 = hSt1	hSt1 = hSt1	hSt1 = hSt1
hNo1 = hNo1	hNo1 = iPh7	hNo1 = sGn2	hNo1 = hNo1
sGa1 = hH1	sGa1 = aIa2	sGa1 = aIa2	sGa1 = aIa2
aIa1 = aIa1	aIa1 = aIa1	aIa1 = aIa1	aIa1 = aIa1
aIa2 = aIa2	aIa2 = sGn2	aIa2 = hEh2	aIa2 = hEh2
hH1 = sGa1	hH1 = sGa1	hH1 = sGa1	hH1 = hH1
hEh1 = hEh1	hEh1 = sGn1	hEh1 = hEh1	hEh1 = hEh1
hEh2 = aN7s1	hEh2 = iPh4	hEh2 = aN7s1	hEh2 = sGa1
13/18	6/18	12/18	13/18

We first evaluate temporal correlations by showing the increased precision achieved through successive collection iterations. As depicted in Figure 4, we correctly correlate 13 of the 18 hardware identifiers after six collection iterations using only the temporal data.

Next, we evaluate the results of temporal (T), spatial (S), temporal and spatial (T/S), and finally a weighted temporal and spatial analysis (T*5/S), using a temporal weight coefficient of five, in Table III. Grey cells indicate correct inferences.

While we expected better results with spatial analysis, the performance is limited by our data that contains many co-located devices that move together. With more locations and longer collections, we expect the ambiguity to resolve.

We now add the TAC-OUI correlation technique to our bipartite matching algorithm, with the results in Table IV. There was a slight improvement with the addition of the OUI analysis and no conclusive change when using a weighted temporal value. Although it appears that we lost granularity on devices iPh4 and iPh5, it is important to note that these devices have the exact same characteristic vectors across all correlation maps — they are the same model using the default configuration, and will therefore have the same results for our OUI, User Agent, and Bonjour analysis. As a result, the algorithm correctly groups all of the identifiers of iPh4 and

TABLE IV: Normal vs. Weighted TAC-OUI Correlation

T/S/O	T*5/S/O
iPh1 = iPh1	iPh1 = iPh1
iPh2 = iPh2	iPh2 = iPh2
iPh3 = iPh3	iPh3 = iPh3
iPh4 = iPh4	iPh4 = iPh5
iPh5 = iPh5	iPh5 = iPh4
iPh6 = iPh6	iPh6 = iPh6
iPh7 = iPh7	iPh7 = iPh7
iPa1 = iPa1	iPa1 = iPa1
sGn1 = sGn1	sGn1 = sGn1
sGn2 = sGn2	sGn2 = sGn2
hSt1 = hSt1	hSt1 = hSt1
hNo1 = hNo1	hNo1 = hNo1
sGa1 = aIa2	sGa1 = aIa2
aIa1 = aIa1	aIa1 = aIa1
aIa2 = sGa1	aIa2 = sGa1
hH1 = hH1	hH1 = hH1
hEh1 = hEh1	hEh1 = hEh1
hEh2 = hEh2	hEh2 = hEh2
16/18	14/18

iPh5, but only finds the correct two pairs within that group with chance – the best that the algorithm could possibly do, given the evidence. In real-life scenarios, continued data collection can resolve instances of equivalent evidence.

B. Advanced Adversary Scenario

The second scenario represents an advanced adversary with the capability to process the data-stream of each protocol. Correlations between devices are made using protocol data analysis techniques as well as the aforementioned temporal, spatial, and TAC-OUI methods. We use the same general ILP, but include $k = 7$ different association matrices, given as C^k where: C^1 is temporal, C^2 is spatial, C^3 is OUI, C^4 is UAProf, C^5 is User-Agent, C^6 is DHCP, and C^7 is Bonjour. To balance the relative influence of each, we introduce a weighting coefficient vector $\mathbf{W}^T = (5, 1, 1, 1, 1, 0.75, 0.75)$. Formally:

$$\text{Maximize} \sum_{k=1}^{c} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij}^{k} W^{k} A_{i,j}$$

$$\text{Subject to} \sum_{i=1}^{m} A_{ij} \leq 1, \sum_{j=1}^{n} A_{ij} \leq 1$$

1) TAC – User-Agent: We compare TAC data to the derived User-Agent. The TAC lookup provides us with the manufacturer and a model of a device, as does the User-Agent. Of note, User-Agents do not appear to delineate between specific iPhone models and thereby reduce the granularity of the correlation in those instances. We do a simple substring matching of the TAC manufacturer to the User-Agent manufacturer. Likewise, we match between derived models. Exact matches receive a score of 10, no match or no data receive a score of 0, and partial matches receive a score between 0 and 10.

2) TAC – UAProf: Similarly, we compare manufacturer and model derived from the TAC and the UAProf profile by performing a substring matching of the UAProf URL and using the same scoring from User-Agent analysis. We hypothesize that the User-Agent and UAProf analysis will be the strongest correlation indicators of the protocol-based analysis techniques due to the granularity inherent in the dataset.

3) DHCP: Both DHCP and Bonjour offer a unique form of granularity. Neither specifically identifies the manufacturer of the device; however, the manufacturer can sometimes be inferred. Additionally, both protocols pair well with the OUI analysis, combining the manufacturer correlation (OUI) and the model correlation (DHCP/Bonjour). While DHCP and Bonjour are less granular than that of User-Agent or UAProf analysis, they are valuable in instances where a User-Agent is not collected or in instances where User-Agent resolution fails to disambiguate. Apple devices typically contain the model information in both DHCP and Bonjour protocols and can be used to compare substring matches of the TAC-derived model. Android devices have a hostname of the format `android_X` or `android-X`, where `X` is a number assigned during operating system (OS) build time. Utilizing the previously obtained OS information from the GSM Arena lookup, we can minimally correlate the IMEI and MAC address pair based on OS compatibility. A correlation score of 10 indicates an exact match, a score of 0 indicates no match or no data, a score of 2 indicates an OS match. To date, we have been unable to identify traffic from non-iOS or Android devices.

TABLE V: Protocol Analysis with TAC-OUI Correlation

UA	UAProf	OUI	Bonjour	DHCP
iPh1 = iPh5	iPh1 = iPh2	iPh1 = iPa1	iPh1 = hEh1	iPh1 = iPh5
iPh2 = iPh2	iPh2 = hEh2	iPh2 = iPh6	iPh2 = iPh7	iPh2 = hEh2
iPh3 = iPh4	iPh3 = hEh1	iPh3 = iPh4	iPh3 = hH1	iPh3 = hH1
iPh4 = iPh7	iPh4 = hNo1	iPh4 = iPh2	iPh4 = iPh2	iPh4 = iPh2
iPh5 = iPh3	iPh5 = sGa1	iPh5 = iPh3	iPh5 = iPh5	iPh5 = iPh7
iPh6 = iPh1	iPh6 = iPh1	iPh6 = iPh1	iPh6 = iPh1	iPh6 = iPh1
iPh7 = iPh6	iPh7 = iPh3	iPh7 = iPf1	iPh7 = iPh6	iPh7 = iPh6
iPa1 = iPa1	iPa1 = sGn2	iPa1 = iPh7	iPa1 = hEh2	iPa1 = iPa1
sGn1 = sGn1	sGn1 = sGn1	sGn1 = sGn1	sGn1 = sGn1	sGn1 = sGn1
sGn2 = sGn2	sGn2 = iPh7	sGn2 = sGn2	sGn2 = sGn2	sGn2 = sGa1
hSt1 = hSt1	hSt1 = aN7s1	hSt1 = hEh2	hSt1 = aN7s1	hSt1 = hEh1
hNo1 = hEh1	hNo1 = iPh5	hNo1 = hH1	hNo1 = sGa1	hNo1 = iPh4
sGa1 = sGa1	sGa1 = iPh6	sGa1 = iPh5	sGa1 = iPh3	sGa1 = iPf1
aIa1 = aN7s1	aIa1 = aIa1	aIa1 = aIa1	aIa1 = aIa1	aIa1 = aIa1
aIa2 = iPf1	aIa2 = iPf1	aIa2 = aIa2	aIa2 = iPf1	aIa2 = aIa2
hH1 = hH1	hH1 = hH1	hH1 = hEh1	hH1 = hNo1	hH1 = hNo1
hEh1 = aIa2	hEh1 = aIa2	hEh1 = hSt1	hEh1 = aIa2	hEh1 = aN7s1
hEh2 = aIa1	hEh2 = iPh4	hEh2 = hNo1	hEh2 = iPh4	hEh2 = sGn2
7/18	3/18	4/18	4/18	4/18

4) Bonjour: Bonjour is enabled by default on Apple devices [18] and often substantiates OUI correlation. We observe that Apple devices utilize both DHCP and Bonjour, and provide the same hostname in each for a single device. The value of utilizing both protocols for correlation is to provide redundancy and improve reliability in cases when one or the other cannot be collected. Note that the user may change DHCP and Bonjour hostnames, thereby removing insight into the device other than revealing its OS.

The following correlation techniques are implemented using the bipartite matching algorithm: TAC-User-Agent (U), TAC-UAProf (X), TAC-Bonjour (M), and TAC-DHCP (B). We first evaluate each protocol to include the previously utilized TAC-OUI technique individually without temporal or spatial analysis, results of which are in Table V. As expected, using only TAC-User-Agent provides some fidelity, but must be combined with other techniques to provide accurate correlations.

Next, we evaluate different combinations of the protocols while adding the temporal and spatial correlation. By adding the User-Agent data, we achieve the highest meaningful precision this data set allows when considering the previous explanation for devices iPh4 and iPh5 (Table VI).

Table VII demonstrates several important findings. First, the combination of OUI with DHCP and Bonjour achieves similar performance to that of the User-Agent. This alternate method proves valuable in instances where we do not or cannot collect the User-agent, or where the User-Agent is not

TABLE VI: Results Incorporating User-Agent Data

T/S/UA	T*5/S/UA
iPh1 = iPh1	iPh1 = iPh1
iPh2 = iPh2	iPh2 = iPh2
iPh3 = iPh3	iPh3 = iPh3
iPh4 = iPh4	iPh4 = iPh5
iPh5 = iPh5	iPh5 = iPh4
iPh6 = iPh6	iPh6 = iPh6
iPh7 = iPh7	iPh7 = iPh7
iPa1 = iPa1	iPa1 = iPa1
sGn1 = sGn1	sGn1 = sGn1
sGn2 = sGn2	sGn2 = sGn2
hSt1 = hSt1	hSt1 = hSt1
hNo1 = hNo1	hNo1 = hNo1
sGa1 = sGa1	sGa1 = sGa1
aIa1 = aIa1	aIa1 = aIa1
aIa2 = hEh2	aIa2 = aIa2
hH1 = hH1	hH1 = hH1
hEh1 = hEh1	hEh1 = hEh1
hEh2 = aN7s1	hEh2 = hEh2
16/18	16/18

TABLE VII: Results Incorporating DHCP and Bonjour

T/S/O/B/M	T*5/S/O/B/M	T*5/S/O/B*.75/M*.75
iPh1 = iPh1	iPh1 = iPh1	iPh1 = iPh1
iPh2 = iPh2	iPh2 = iPh2	iPh2 = iPh2
iPh3 = iPh3	iPh3 = iPh3	iPh3 = iPh3
iPh4 = iPh5	iPh4 = iPh4	iPh4 = iPh4
iPh5 = iPh4	iPh5 = iPh5	iPh5 = iPh5
iPh6 = iPh6	iPh6 = iPh6	iPh6 = iPh6
iPh7 = iPh7	iPh7 = iPh7	iPh7 = iPh7
iPa1 = iPa1	iPa1 = iPa1	iPa1 = iPa1
sGn1 = sGn1	sGn1 = sGn1	sGn1 = sGn1
sGn2 = sGn2	sGn2 = sGn2	sGn2 = sGn2
hSt1 = hSt1	hSt1 = hSt1	hSt1 = hSt1
hNo1 = hEh2	hNo1 = hEh2	hNo1 = hNo1
sGa1 = sGa1	sGa1 = sGa1	sGa1 = sGa1
aIa1 = aIa1	aIa1 = aIa1	aIa1 = aIa1
aIa2 = aIa2	aIa2 = aIa2	aIa2 = aIa2
hH1 = hH1	hH1 = hH1	hH1 = hH1
hEh1 = hEh1	hEh1 = hEh1	hEh1 = hEh1
hEh2 = hNo1	hEh2 = hNo1	hEh2 = hEh2
14/18	16/18	18/18

discriminatory. Additionally, we found that by lowering the relative weight of DHCP and Bonjour (to 0.75), we ensure they do not overwhelm stronger sources of evidence.

It is difficult to interpret our UAProf results as only three of our devices transmitted a UAProf URL. Only one device provided data that could be correlated to the TAC (hH1 - HTC Hero), as the manufacturer for the two aIa devices (Acer Iconia A501) was different than the manufacturer derived from the TAC. We can see in Table VIII that while previous temporal and spatial analysis did not correctly correlate the identifiers for device hH1, we were able to correctly correlate the device using its UAProf data. Devices iPh4 and iPh5 were seen to have changed again, due to their inherent similarity. We believe that a larger device dataset including more devices sending UAProf will enhance the utility of this form of correlation. We have observed a significant number of mobile devices utilize the UAProf x-wap-profile in our continued research — 45 out of 134 devices in our dataset.

The test results in Table IX show the effects of using all of our techniques combined with weighted coefficients for the temporal, Bonjour, and DHCP datasets. Using these coefficients, we achieve the highest accuracy possible using the complete correlation technique.

C. Leaked IMEI

Lastly, we examine a very strong form of correlation: IMEIs leaked in the payload of 802.11 traffic. While such

TABLE VIII: Results After Incorporating UAProf Data

T/S/UAProf	T*5/S/UAProf
iPh1 = aIa1	iPh1 = aIa1
iPh2 = iPh2	iPh2 = iPh2
iPh3 = iPh3	iPh3 = iPh3
iPh4 = iPh4	iPh4 = iPh5
iPh5 = iPh5	iPh5 = iPh4
iPh6 = iPh6	iPh6 = iPh6
iPh7 = iPh7	iPh7 = iPh7
iPa1 = iPa1	iPa1 = iPa1
sGn1 = sGn1	sGn1 = sGn1
sGn2 = sGn2	sGn2 = sGn2
hSt1 = hSt1	hSt1 = hSt1
hNo1 = hNo1	hNo1 = hNo1
sGa1 = sGa1	sGa1 = aIa2
aIa1 = iPh1	aIa1 = iPh1
aIa2 = aIa2	aIa2 = sGa1
hH1 = hH1	hH1 = hH1
hEh1 = hEh1	hEh1 = hEh1
hEh2 = hEh2	hEh2 = aN7s1
16/18	12/18

TABLE IX: Results After Incorporating All Collected Data

T/S/O/U/X/M/B	T*5/S/O/U/X/M/B	T*5/S/O/U/X/M/B*.75	T*5/S/O/U/X/M*.75/B*.75
iPh1 = iPh1	iPh1 = iPh1	iPh1 = iPh1	iPh1 = iPh1
iPh2 = iPh2	iPh2 = iPh2	iPh2 = iPh2	iPh2 = iPh2
iPh3 = iPh3	iPh3 = iPh3	iPh3 = iPh3	iPh3 = iPh3
iPh4 = iPh5	iPh4 = iPh5	iPh4 = iPh5	iPh4 = iPh4
iPh5 = iPh4	iPh5 = iPh4	iPh5 = iPh4	iPh5 = iPh5
iPh6 = iPh6	iPh6 = iPh6	iPh6 = iPh6	iPh6 = iPh6
iPh7 = iPh7	iPh7 = iPh7	iPh7 = iPh7	iPh7 = iPh7
iPa1 = iPa1	iPa1 = iPa1	iPa1 = iPa1	iPa1 = iPa1
sGn1 = sGn1	sGn1 = sGn1	sGn1 = sGn1	sGn1 = sGn1
sGn2 = sGn2	sGn2 = sGn2	sGn2 = sGn2	sGn2 = sGn2
hSt1 = hSt1	hSt1 = hSt1	hSt1 = hSt1	hSt1 = hSt1
hNo1 = hEh2	hNo1 = hEh2	hNo1 = hNo1	hNo1 = hNo1
sGa1 = sGa1	sGa1 = sGa1	sGa1 = sGa1	sGa1 = sGa1
aIa1 = aIa1	aIa1 = aIa1	aIa1 = aIa1	aIa1 = aIa1
aIa2 = aIa2	aIa2 = aIa2	aIa2 = aIa2	aIa2 = aIa2
hH1 = hH1	hH1 = hH1	hH1 = hH1	hH1 = hH1
hEh1 = hEh1	hEh1 = hEh1	hEh1 = hEh1	hEh1 = hEh1
hEh2 = hNo1	hEh2 = hNo1	hEh2 = hEh2	hEh2 = hEh2
14/18	14/18	16/18	18/18

leakage violates typical abstraction barriers, [1], [8], [9], [10], [11] reference approximately 900 applications that use a device's IMEI, or the hashed IMEI, to provide a unique pseudo-anonymous identifier for tracking and advertisements. We improve on the methods identified by [11]; instead of creating hash tables of all possible IMEIs, we instead use temporal analysis to target specific IMEIs.

It is trivial to correlate an IMEI and a MAC address if we can collect the IMEI from an application over 802.11. We empirically test 16 mobile applications previously identified to leak the IMEI on Android, iOS, and Windows Phone 7 operating systems [1], [9], [10], listed in Table X. Eight of the sixteen Android applications leaked either the IMEI or hashed IMEI. The remaining eight applications either did not leak the identifiers or we failed to detect them in the traffic. We found no iOS or Windows Phone applications that leak their IMEI due to restrictions in place by both platforms that prevent the application from accessing the IMEI [19], [20]. Although tangential, we note that three iOS applications leak the Unique Device Identifier (UDID).

D. Future Work

We intend to further our work by integrating 802.15 analysis techniques. We plan to study both pairwise (802.15-GSM, 802.15-802.11) correlation and methods of correlating all three identifiers. Future work in 802.11 and 802.15 identifier correlation will borrow from and enhance "Wi-Fi and Bluetooth MAC Address One-Off" techniques [21].

TABLE X: Applications Leaking Device Identifiers

Android	Leaks	iPhone	Leaks	Windows	Leaks
AutoRun		n/a		n/a	
Assistant	SHA1	Assistant		Assistant	
Classic Simon		n/a		n/a	
Documents To Go 3.0		n/a		n/a	
Droid Jump		n/a		n/a	
iHeartRadio	IMEI	iHeartRadio	UDID	iHeartRadio	
KAYAK		KAYAK		n/a	
Moco Chat, Meet, Games	IMEI	Moco Chat, Meet, Games		MocoSpace	
Moron Test: Old School	IMEI	Moron Test: Old School	UDID	n/a	
Moron Test: Section 2		n/a		n/a	
Paper Toss		Paper Toss		n/a	
Smart Simon	MD5, SHA1	n/a		n/a	
SmartTacToe		SmartTacToe	UDID	n/a	
Starbucks		Starbucks		n/a	
Video Poker	MD5, SHA1	n/a		n/a	
Video Poker		n/a		n/a	
White & Yellow Pages	IMEI	White & Yellow Pages		White & Yellow Pages	
Yellow Pages	IMEI, MD5	Yellow Pages		Yellow Pages	

Future efforts should expand the breadth and scope of mobile devices by collecting and analyzing CDMA identifiers, such as the Electronic Serial Number and Mobile Equipment Identifier (ESN, MEID). Similar to the GSM IMEI, the ESN and MEID reserve bits that identify the manufacturer and serial number, which may correlate CDMA devices [22].

Continued work should investigate a more principled weighting algorithm, as opposed to the empirically derived weights in this work. Better weight assignment may provide more accurate correlation by overcoming instances where less granular data conflicts with more granular correlation.

Last, we plan to investigate the application of feature vector similarity matching to address the issues observed with iPh4 and iPh5 in §III-A.

IV. Appendix

To demonstrate the feasibility and power of our framework in associating disparate hardware identifiers, we employed simulated temporal and spatial data. In this subsection, we illustrate that such collection is feasible.

A GSM device performs cell selection upon power-on, monitoring Broadcast Control Channel (BCCH) messages to determine the optimal radio frequency (RF) connection [23]. The mobile handset selects the cell with the highest C1 value, where C1 is defined by 3GPP to be a function of device and tower transmit power [24].

We wish to induce mobile devices to reselect from their current cell tower to a rogue tower under our control. We focus on cell reselection when the mobile device is in idle mode to avoid complications with handovers and call routing when a handset is engaged in an active phone call. In idle mode, a mobile device will monitor BCCH parameters and associated RF measurements of the relevant towers, to choose a cell according to the C2 algorithm [23]. The Cell Reselection Offset (CRO) is normally utilized by the network to handle load balancing and encourages reselection to nearby towers even if they are not the best RF path. A cell operates with a $CRO = 0$ in typical scenarios, and thus C2 is calculated with the same parameters as C1 [24].

$$C2 = C1 + CRO - (TempOffset * PenaltyTime)$$

In order to force the mobile phone to reselect to our rogue tower, we transmit an abnormally high CRO on our BCCH to induce reselection to our rogue BTS. OpenBTS [25] is one possible implementation of a rogue BTS that can collect device IMEIs. With the release of C2.8, OpenBTS supports the ability to delineate the CRO setting. As each mobile device registers with our tower, one can record the IMEI and then release the mobile device back to the local network [2], [26].

References

[1] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones," in *Proceedings of the 9th USENIX OSDI conference*, 2010, pp. 1–6.

[2] R.-P. Weinmann, "Baseband attacks: Remote exploitation of memory corruptions in cellular protocol stacks," in *USENIX Workshop on Offensive Technologies (WOOT12)*, 2012.

[3] C. Mulliner, N. Golde, and J.-P. Seifert, "SMS of Death: from analyzing to attacking mobile phones on a large scale," in *Proceedings of the 20th USENIX conference on Security*, 2011, pp. 24–24.

[4] K. Nohl, "Rooting SIM Cards," in *Blackhat Conference*, 2013.

[5] D. Eastlake 3rd, "IANA Considerations and IETF Protocol Usage for IEEE 802 Parameters," RFC 5342, IETF, Sep. 2008.

[6] G. Association, "IMEI Allocation and Approval Guidelines Version 6.0," Jul. 2011.

[7] S. L. Garfinkel, A. Juels, and R. Pappu, "RFID Privacy: An Overview of Problems and Proposed Solutions," *Published by the IEEE Computer Society*, p. 14, May 2005, http://www.cs.colorado.edu/~rhan/CSCI_7143_Fall_2007/Papers/rfid_security_01439500.pdf.

[8] M. Egele, C. Kruegel, E. Kirda, and G. Vigna, "PiOS: Detecting privacy leaks in iOS applications," in *Proceedings of the Network and Distributed System Security Symposium*, 2011.

[9] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall, "These aren't the droids you're looking for: retrofitting android to protect data from imperious applications," in *Proceedings of the 18th ACM CCS conference*, 2011, pp. 639–652.

[10] G. Eisenhaur, M. N. Gagnon, T. Demir, and N. Daswani, "Mobile Malware Madness and How to Cap the Mad Hatters," in *Blackhat Conference*, 2011.

[11] M. N. Gagnon, "Hashing IMEI numbers does not protect privacy," Dasient Blog, 2011, http://blog.dasient.com/2011/07/hashing-imei-numbers-does-not-protect.html.

[12] "GSM Association TS.06 - IMEI Address and Approval Guidelines, Version 6.0," http://www.gsma.com/newsroom/wp-content/uploads/2012/06/TS.06-v6.0.pdf.

[13] T. Rappaport, *Wireless Communications, Principles and Practice. Second Edition.* Upper Saddle River, NJ: Prentice Hall, 2002.

[14] "ETSI TS 122 016 V9.0.1 (2010-04) Digital cellular telecommunications system (Phase 2+); UMTS; LTE; IMEI (3GPP TS 22.016 version 9.0.1 Release 9)," http://www.etsi.org/deliver/etsi_ts/122000_122099/122016/09.00.01_60/ts_122016v090001p.pdf.

[15] IEEE, "OUI Public Listing," http://standards.ieee.org/develop/regauth/oui/oui.txt.

[16] "Wireless Universal Resource FiLe," http://wurfl.sourceforge.net.

[17] "GNU Linear Programming Kit, Version 4.48," 2013, http://www.gnu.org/software/glpk/glpk.html.

[18] "Avahi mDNS/DNS-SD daemon," 2013, http://manpages.ubuntu.com/manpages/precise/man8/avahi-daemon.8.html.

[19] "How to get IMEI on iPhone?" http://stackoverflow.com/questions/823181/how-to-get-imei-on-iphone.

[20] "How to get IMEI," http://social.msdn.microsoft.com/Forums/en-US/wpdevelop/thread/64ba34b4-552e-44b7-9927-0e32144ea11a/.

[21] J. Cache and V. Liu, *Hacking Wireless Exposed.* McGraw-Hill, 2007.

[22] "CDMA Hardware Identifiers - ESN, MEID, pESN," https://sites.google.com/site/bbayles/index/cdma_hardware_id.

[23] "GSM/GPRS Radio Network Planning," http://www.ehanworld.com/GSM/GSM.html.

[24] 3rd Generation Partnership Project (3GPP), "Technical Specification Group GSM/Edge; Radio subsystem link control," 1999.

[25] D. A. Burgess, H. S. Samra *et al.*, "The OpenBTS Project," 2008, http://openbts.sourceforge.net.

[26] D. Strobel, "IMSI Catcher," *Chair for Communication Security, Ruhr-Universität Bochum*, p. 14, 2007.