Real Threats to Your Data Bills: Security Loopholes and Defenses in Mobile Data Charging

Chunyi Peng
The Ohio State University
chunyi@cse.ohio-state.edu

Chi-Yu Li, Hongyi Wang, Guan-Hua Tu, Songwu Lu
University of California, Los Angeles
{lichiyu,hywang,ghtu,slu}@cs.ucla.edu

ABSTRACT

Secure mobile data charging (MDC) is critical to cellular network operations. It must charge the *right* user for the *right* volume that (s)he authorizes to consume (*i.e.*, requirements of authentication, authorization, and accounting (AAA)). In this work, we conduct security analysis of the MDC system in cellular networks. We find that all three can be breached in both design and practice, and identify three concrete vulnerabilities: *authentication bypass*, *authorization fraud* and *accounting volume inaccuracy*. The root causes lie in technology fundamentals of cellular networks and the Internet IP design, as well as imprudent implementations. We devise three showcase attacks to demonstrate that, even simple attacks can easily penetrate the operational 3G/4G cellular networks. We further propose and evaluate defense solutions.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

Keywords

Cellular Networks; Mobile Data Services; Authentication; Authorization; Accounting; AAA; Attack; Defense

1. INTRODUCTION

Mobile data services are getting increasingly popular, thanks to the proliferation of smartphones and tablets, as well as the rapid deployment of the third-generation/fourth-generation (3G/4G) cellular networks. Global mobile data traffic grew 81% in 2013 and is projected to increase 11-fold in the following five-year span [16]. This is contributed by 2.1 billion mobile Internet users worldwide (by 2013 June), including 299 million 3G/4G broadband subscribers (95% of inhabitants) in the US [25].

Convenient mobile data access does come with cost for users. Most cellular operators charge mobile users based on their consumed data volume [8, 32]. Mobile users pay for the data usage at a preset price within certain volume cap, or in the pay-per-use manner. For example, AT&T, a Tier-1 US carrier, charges $20 for 300MB per month for domestic access and about 2¢ per KB during international travels [10]. This *volume-based charging* scheme is not adopted without rationale. The radio spectrum is scarce and expensive (in spectrum licensing [20]), and wireless speed is bounded by Shannon channel capacity. The explosive growth of mobile data traffic further justifies the metered charging.

Undoubtedly, a well-designed and properly-operated mobile data charging (MDC) system is critical to cellular networks. It not only safeguards the multi-trillion revenue of global operators, but also protects the monetary rights of billions of mobile users. To enable metered charging, the key is to collect how much data is *actually* used by which mobile user when (s)he agrees to. A secure MDC should meet three requirements:

1. **[Authentication]** *The user being billed for the given data transfer must be the one who actually does the transfer.* MDC must authenticate the user who consumes the actual data usage.
2. **[Authorization]** *The data usage and its associated charge should be with the user's consent.* A user should only pay for those authorized data services, but not the spam from attackers.
3. **[Accounting volume]** *The volume should be accurate.* The recorded volume should be identical to that transferred at the user device.

At first glance, it appears straightforward to meet the above requirements. The MDC method is officially stipulated by the 3GPP specification [3]. It is performed inside the cellular core network. Whenever a data session is initiated with the mobile device, the traffic from/to the mobile device traverses the cellular gateways (akin to edge routers or switches in the Internet) to reach the destination. The gateway counts the payload of *observed* data packets for each mobile data session as the volume. It further associates this volume with the user who initializes and uses this data session. Given user authentication, the network is capable of inferring who uses this data session. To prevent unauthorized access, cellular operators also deploy firewalls and network address translators (NATs) to shield mobile devices from the traffic types of no interest. While recent studies [21, 22, 27, 28, 36] have reported various cases on accounting volume inaccuracy, the two aspects of authentication and authorization still look bullet-proof. They seem almost impossible to go wrong. Anyway, authentication and authorization have been well studied in the security community, and their solutions to cellular networks have been generally successful to date.

However, MDC is not as secure as anticipated. We discover that, it is also vulnerable in authentication and authorization. Charging actions may be taken upon the wrong user, or on data that the cellu-

CCS'14, November 3–7, 2014, Scottsdale, Arizona, USA.
Copyright 2014 ACM 978-1-4503-2957-6/14/11 ...$15.00.
http://dx.doi.org/10.1145/2660267.2660346 .

Dimension	Threat	Loophole	Attack	Root Cause	Defense solution
Authentication(§3.1)	A user is billed for other's data traffic.	Authentication Bypass	Free-uplink-attack via IP spoofing	No cross-layer secure binding (§3.4)	Cross-layer secure binding in the data plane(§4.1)
Authorization(§3.2)	Unwanted data is allowed and billed.	Authorization Fraud	Cloak-and-dagger attacks via MMS and IP spoofing	Network-based authorization; IP push model (§3.4)	Explicit de-authorization on demand in the control plane(§4.2)
Accounting volume(§3.3)	A user is billed for data never received.	Accounting Vol. Inaccuracy	Hit-but-no-touch TTL-based attack	open-loop accounting; Independent PS delivery(§3.4)	Feedback from the end/network; explicit de-authorization(§4.2,§4.3)

Table 1: Summary of results.

lar user does not want. Consequently, all AAA components can be breached in both technology and practice. The deployed defense measures fail to protect them. Specifically, we identify three loopholes: *authentication bypass*, *authorization fraud*, and *accounting volume inaccuracy*. All threats may impose real monetary loss to the victim user(s).

Moreover, to our surprise, no sophisticated attack models are needed. Simple attacks may work in operational 3G/4G cellular networks! By significantly limiting the capability of the adversary and applying variations of well-known attack methods (*e.g.*, IP spoofing), we have devised a few showcase attacks to test all three AAA dimensions. All can pass the defense measures deployed by cellular operators. The attacks can be against an individual or a group of victims of any size, without requiring control or access to the victim phone or the carrier. Our experiments further indicate that the incurred charging damage exhibits no sign of limit.

We further analyze their root causes. They are beyond our initial thoughts of being induced by implementation bugs from vendors and imprudent practice by operators. Factors rooted in the technology basics stipulated in the 3G/4G standards also share the responsibility. They include lack of cross-layer secure binding in authentication, network-based decision in authorization, and open-loop operation and IP push delivery model in accounting. To fix these loopholes, we further propose defense solutions, which apply three guidelines of cross-layer security binding, coordinated control-plane and data-plane operations for security, and infrastructure-assisted end-user feedback.

Table 1 summarizes our main results. Before we delve into the details, we rush to clarify what this work is not about. We examine the technical side of MDC security, but have not looked into the issue of attack incentives. We focus on how (rather than why) adversaries attack the system. While our proposed solution offers one feasible approach, other alternatives (*e.g.*, deterrence by detecting and punishing attackers) are also possible as elaborated later. Specifically, we make four following contributions:

- We uncover security threats in the MDC system, and confirm that mobile users are vulnerable to unconstrained monetary loss.
- We expose and validate security loopholes in all three AAA dimensions. We show that all AAA components can be breached in both technology and operations. The deployed security measures fail to protect them.
- We further sketch novel attacks that exploit such loopholes and validate them through experiments in operational 3G/4G cellular networks.
- We deduce direct causes rooted in both the technology basics and imprudent practice by operators. We propose and evaluate defense solutions.

The rest of the paper is organized as follows. §2 introduces the MDC background, the threat model and the experimental methodology. §3 elaborates our security analysis on each AAA dimension. §4 proposes defense solutions and §5 evaluates them. §6 compares with related work and §7 concludes the paper.

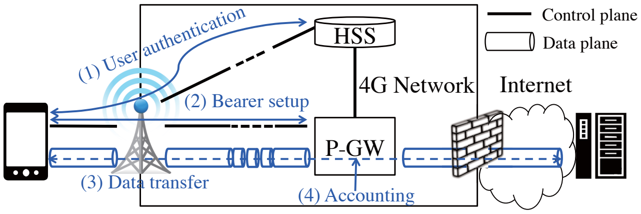


Figure 1: Main operations for mobile data transfer and charging in 4G LTE networks.

§3 elaborate our security analysis on each AAA dimension. §4 proposes defense solutions and §5 evaluates them. §6 compares with related work and §7 concludes the paper.

2. MOBILE DATA CHARGING IN 4G/3G

Figure 1 illustrates major operations for mobile data delivery and its charging in cellular networks. We use 4G Long Term Evolution (LTE) as the default network setting. The case for 3G networks is similar. Specifically, there are four main steps. The first step is to verify whether the user equipment (UE) is legitimate to use cellular networks. It is mandatory except for dialing 911 calls. This is done through user authentication when the UE initially attaches to cellular networks (*e.g.*, the phone powers on). Second, the authenticated UE establishes data bearers for subsequent transfer. It is a prerequisite to obtain granted data access (*i.e.*, IP connectivity) from the cellular carrier before running data services.

Afterwards, it is ready to start any data service (*e.g.*, web browsing and video streaming) (Step 3). The data packets are delivered from the UE to the base station, and then forwarded to the gateway (*i.e.*, P-GW in LTE), and finally to the external host, or vice verse. At the border to the Internet, cellular operators deploy border gateways and middleboxes, including firewalls and NATs [5, 38].

Volume accounting (Step 4) is performed in parallel with data transfer. Usage volume is collected when data packets traverse the gateway along both inbound (*i.e.*, phone-destined) and outbound (*i.e.*, phone-originated) directions. To infer who should pay the bill, the gateway uses a unique charging ID on a per-flow basis, or on a per-IP basis. Each charging ID is correlated with a registered user via the first two steps, as elaborated later.

2.1 Threat Model

We expose security vulnerabilities of the MDC system without giving the adversary too much attack power. This is done by assuming that all other components in the cellular networks and mobile phone victims are not compromised and via limiting the exploits to be used by the adversary.

Specifically, the adversary can be a mobile user or a static host on the Internet, whereas the victim is typically another mobile user and loses money due to attacks from the adversary. The victim user can be chosen on purpose (given a specific phone number) or at random. In some scenarios, the operator might become the victim because the attack affects all users and degrades the overall performance. The adversary has no access to the cellular network infrastructure or other devices. It solely exploits the public available information when launching attacks. The adversary only has full control over its own smartphone and a remote server. The phone is a programmable commodity smartphone, *e.g.*, an Android phone. The server is a commodity one (without super-powerful computation or communication capabilities), deployed outside the cellular network. It is also programmable, and can use any available tricks over the Internet (*e.g.*, using Tor [2] to hide its identity and location) to cheat the cellular carrier.

We further assume that all other mechanisms in cellular networks and at other mobile clients work properly. Therefore, the attacker cannot leverage improper operations in other components to launch attacks against MDC. All components function normally without any compromise, misconfiguration, malware, or intrusion.

In reality, the adversary could be more damaging since exploits in other components are possible, such as SIM/USIM card hacking [1], authentication protocol vulnerability [9, 26], firewalls misconfiguration [29, 30], mobile malwares (see [11, 13, 17, 37, 39]), *etc.*. For example, the malware installed on the victim phone may permit the attacker to do what he wants (*e.g.*, keep sending junk data). It does make it easier to launch an attack. However, MDC may not hold the main responsibility since service requests do come from the devices. Instead, we focus on a modest threat model to show that the exploits in MDC vulnerabilities are readily accessible. Given this model, the identified security loopholes may translate into realistic attacks, thus exposing practical threats to operational 3G/4G infrastructures and mobile users. Giving more power to the adversary only aggravates the incurred damage.

2.2 Experimental Methodology

To validate security loopholes and assess their damage in operational networks, we design experiments in two major US carriers (OP-I and OP-II) that together cover more than 50% of US subscribers. We run tests at various locations in five US states on the west coast, east coast, and midwest. We also assess various network technologies (4G/3G/2.5G) supported by both carriers. We use several Android phone models, including Samsung S4/S3/S2/Note, HTC one and LG Optimus E970. Since we have no access to the internal cellular infrastructure, we learn their operations from the standard specification and experimental observations. For mobile data transfer and its charging, we collect traces from the phones and our deployed server.

Our experiments are designed to be responsible. We realize that some proposed exploits and their verification tests might be detrimental to operators or other users. Our test was thus conducted with several guidelines: (1) Actual data usage is kept below the data plan cap, regardless of being charged or not; (2) Attacks are performed by using our own phone as the victim; (3) Verification experiments are restricted via small-scale sampling to confirm vulnerabilities in real networks. No large-scale tests are performed.

3. SECURITY ANALYSIS OF MDC

We now examine each individual security element in MDC. Given each element, we analyze its current solution, identify its security loopholes, deduce its causes, sketch showcase attacks, and validate them in operational 3G/4G networks. Note that our main goal is to identify vulnerabilities in MDC. The devised attacks simply illustrate how easy it is to use known attack techniques to breach the MDC system. Moreover, large-scale attacks are feasible, *e.g.*, by exploiting Botnets or using multiple malicious servers. They can be launched from the Internet, which is beyond control of cellular operators.

3.1 On Authentication

3.1.1 Current Solution

To ensure authentication, the current 3G/4G networks have adopted mechanisms at multiple layers of the protocol stack. It adopts user authentication (Step 1) and IP address authentication (Step 2), which are performed during the initial attach procedure.

Figure 2(a) depicts the attach procedure. The baseline user authentication (Step 1) is ensured through the Authentication and Key Agreement (AKA) procedure [7]. Each user obtains a unique and permanent ID, called international mobile subscriber identity (IMSI). The confidential IMSI and its related key for user authentication are securely stored in both the SIM/USIM card at the user side and the Home Subscriber Server (HSS, akin to a database) at the operator side. When the phone initially attaches to cellular networks, AKA uses challenge-response based mechanisms to verify whether its local IMSI matches with the record stored in the database. A temporary identity derived from IMSI is then used to set up a secure connection against eavesdropping. Once completed, IP address authentication (Step 2) is performed through this secure connection during the bearer activation process. A bearer is for subsequent data transfer. Specifically, the Evolved Packet System (EPS) bearer is established to enable the connection-oriented transmission in the 4G network. It is further carried by an underlying GTP-U (GPRS Tunneling Protocol-User Plane) tunnel. During this process, an IP address is allocated by the gateway, to the UE through this secure connection. Consequently, the IP address is authenticated with the UE.

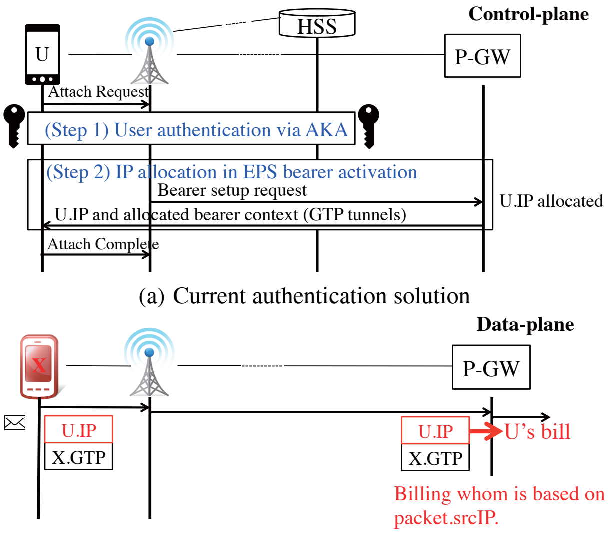
Such IP address authentication is *mandatory* in cellular networks. This is a key difference from the Internet, where such authentication is rarely required. From the charging standpoint, MDC is thus able to map the charging (via the packet header, *e.g.*, IP address) into the authentic user.

3.1.2 Vulnerability Analysis

We discover a loophole that allows for bypassing the above authentication scheme. The root cause lies in neither secure cross-layer binding nor coordination between control and data planes.

As described above, cellular networks indeed perform *control-plane* authentication when assigning an IP address. However, for packet delivery on the *data plane*, enforcement of the assigned, authentic IP address may be missing. The prior authentication is circumvented when a forged IP address is embedded in the data packet. MDC further associates its charging *only* based on the packet header. Moreover, the current solution lacks secure cross-layer binding. In cellular networks, data communication spans *multiple* layers of the protocol stack. A transport-layer flow uses IP packet delivery (Layer 3, L3), which is further carried by GTP-U tunnels (Layer 2, L2). In Step 2, a tunnel ID (that identifies the GTP-U tunnel) is created by the core gateway and made known to other gateways. Although data delivery to/from the UE is only allowed over authenticated L2 tunnel, the L3 IP address carried by the GTP-U payload is not required for verification. This no-binding operation results in an *authentication-bypass* loophole for the charging process which is based on the IP address. For example, as shown in Figure 2(b), when an adversary `X` forges `U`'s IP address in his data transfer, MDC might charge `U` but not `X`.

Note that authentication is critical to both upstream and downstream packets. However, authentication bypass vulnerability may not take effects on downstream packets unless the phone does not


Control-plane
U
HSS
P-GW
Attach Request
(Step 1) User authentication via AKA
(Step 2) IP allocation in EPS bearer activation
Bearer setup request
U.IP allocated
U.IP and allocated bearer context (GTP tunnels)
Attach Complete
Data-plane
X
P-GW
U.IP
X.GTP
U.IP
X.GTP
U's bill
Billing whom is based on packet.srcIP.

(a) Current authentication solution

(b) Authentication bypass loophole in MDC

Figure 2: Current authentication solution and the authentication bypass loophole for MDC on the data plane.

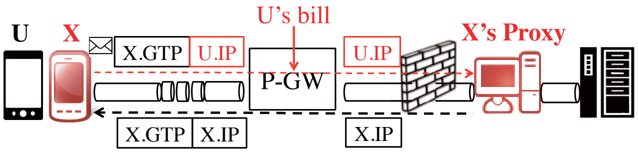


Figure 3: Illustration of Free-Uplink-Attack. Attacker X exploits the authentication bypass loophole.

intend to receive the real data. The forwarding path for downstream is determined by the gateway based on the IP address and the predefined mapping. If authentication goes wrong, downstream packets arrive at the wrong destination. We thus focus on upstream only.

3.1.3 Exploit: Free Uplink Attack

The above loophole can be readily exploited for free uplink access. Figure 3 illustrates the main attack idea. It uses IP spoofing and shifts one's uplink traffic cost to other users. To facilitate bidirectional data delivery, we deploy a proxy outside cellular networks, which helps to forward downlink traffic to the true sender when needed. The attack works as follows. First, the UE probes several forged IP addresses and finds the spoofable ones. We find that not every IP spoofing works in the following validation. Second, the UE registers its genuine IP address with the proxy if it wants to receive the downlink traffic. This is used to traverse NATs and firewalls; the UE must send a packet first to the proxy to allow the downlink traffic in. Finally, the UE uses the forged IP address to deliver its uplink data transfer through the tunnel to the destination via the proxy. For UDP, the proxy only needs to modify the IP address and the checksum in the packets. For TCP, the proxy uses the Split-TCP scheme [12] to split an end-to-end TCP connection into two separate ones. Everything else runs as usual, except uplink packets are delivered with a forged IP address. This attack is appealing to those applications with heavy outbound traffic, such as phone backup, file upload (*e.g.*, photo and video), and video instant messaging (*e.g.*, Skype, FaceTime). Although not all applications carry heavy outbound traffic, free of uplink charges still stimulates such an exploit. More importantly, such a simple attack illustrates the vulnerabilities in operational MDC systems.

Property		**OP-I**	**OP-II**
IP spoofing is feasible		√	√
Maximum spoofable MSB		√ (24)	√ (32)
Fully spoofable?	max	√ (only in 3G)	√
(spoofable ratio)	median	× (10 − 16%)	√ ($m \leq 21$)
(spoofable ratio)	large m	× (<0.4%,m>16)	× (< 1%, m>21)
MDC based on IP addr.		√	×

Table 2: Summary of IP spoofing in two US carriers.

IP spoofing is a well-known, yet unsolved security threat to the wired Internet [14]. However, different from the Internet, cellular MDC is still vulnerable even with user authentication! IP spoofing was observed in some cellular carriers [38], but its impact on MDC has not been examined. In fact, our in-depth study shows that not each IP spoofing turns into real charging threats or even succeeds.

3.1.4 Experimental Validation

We carry out experiments to validate its vulnerability in two US operational networks and assess the damage of the sketched attack. Our empirical study confirms that (i) authentication loophole indeed exists in operational networks. The carrier charges the wrong user who does not perform data delivery. We find out that, IP address can be spoofed in both carriers and charging based on the IP address is used in OP-I. In the tested attack, we confirm that (ii) the malicious phone gains free uplink access. There is no sign of free volume limit. We further uncover that (iii) not each IP spoofing succeeds. It is constrained by geographic locations, cellular technologies (*e.g.*, 4G and 3G), and policy enforcement.

Loophole verification. We address two issues using experiments: (1) Is IP spoofing feasible in operational networks? (2) How does it affect user charging? Though IP spoofing is observed in [38], details are not given and Issue (2) is unaddressed. We thus conduct two experiments. We explore the feasibility by sending data packets from our phone to our server deployed outside the cellular carrier, using various fabricated source IP addresses via `Raw Socket` programming. To answer the second question, we run the following two-phone experiment. Phone `X` uploads 1MB UDP traffic to our server using the IP address of phone `U`. During the test, all other data services and background traffic are cleaned up. We compare the volume sent by `X`, and the itemized billing records for `X` and `U` from the operator. Once `U` is charged, billing must be based on the source IP address. Otherwise, if `X` is charged, spoofing does not threaten data billing.

Table 2 summarizes our results in both carriers. We also plot the spoofing results in both carriers and those in OP-I only using 2.5G/3G/4G technologies in Figures 4 and 5, respectively. We make three observations. First, both carriers allow IP spoofing, and the spoofable most significant bit (MSB) is large. OP-I allows up to /24 in spoofing (covering all the Class-A private address block (10.x.x.x)), whereas OP-II allows up to /32. Second, not all IP addresses are forgeable, and spoofable ratios fluctuate in both carriers. In OP-I, the median spoofing ratio is around 10–16% when the spoofable MSB is smaller than 16 ($m \leq 16$). It shrinks sharply (<0.4%) when m>16. In contrast, OP-II has much larger spoofable ratio. In most cases, even the entire /21 block is spoofable. It becomes low when m is large (m≥21). Third, spoofable ratios are correlated with the used technologies (2.5G/3G/4G). In OP-I, its 3G network is the easiest to be spoofed. Full spoofability when $m \leq 16$ is observed in 3G; this occurs when certain external gateways are used. We gauge that all these are caused by the different policies on IP address allocation and filtering.

To infer the charging constraints, we conduct the two-phone experiments in various scenarios via OP-I. We find that both phones

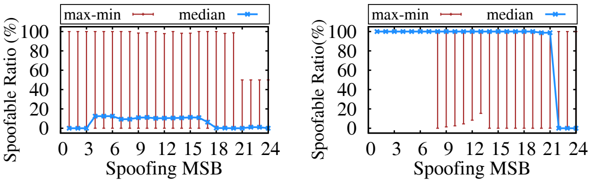


Figure 4: Spoofable ratios in OP-I (left) & OP-II (right).

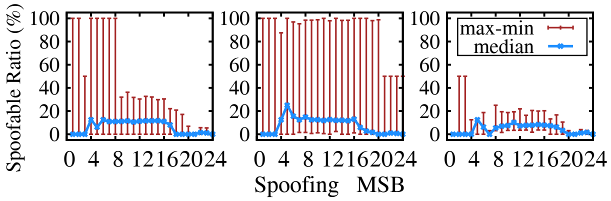


Figure 5: Spoofable ratios in OP-I's networks: 4G LTE (left), 3G UMTS (middle), 2.5G GPRS (right).

are not billed when using different technologies (*e.g.*, 4G and 3G) and being placed across locations (*e.g.*, one in east coast, the other in west coast). We gauge that IP address blocks might be reused across geographic regions or by different technologies. The forged address by X might be reused by another UE, not U. However, it will not affect the free uplink access (X is still free of charge).

Attack assessment. We assess the threats of the proposed free uplink attack. We examine the volume limit by varying it from 1MB to 100MB. All packets are delivered without charge. No sign of volume limits is observed.

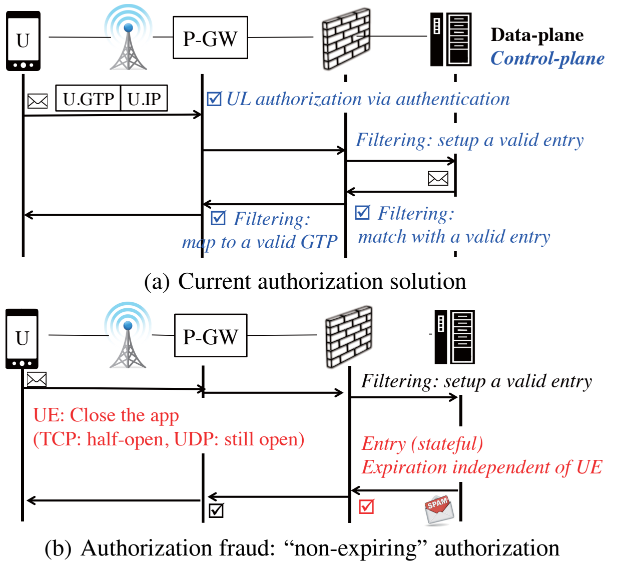
3.2 On Authorization

3.2.1 Current Solution

The authorization for MDC concerns charging actions with or without *user content*. This is slightly different from the Internet case, where authorization is performed by an ISP to let certain traffic pass through. In 3G/4G networks, it varies for two types of data transfer: inbound and outbound.

The outbound transfer is authorized through *implicit* user consent based on authentication. To initiate data service, the UE must be authenticated first. Afterwards, packets from the authenticated UE are sufficient to signify that the UE authorizes the data transfer and its charging.

The case is different for the inbound data transfer, where the UE is at the last hop to receive data and charging is already performed upstream. Three mechanisms are used to ensure implicit authorization. First, deployed firewalls and NATs help. Firewalls prevent traffic types of no interest from getting into the network, while NATs isolate the cellular networks from the external, public Internet via private IP address and port mapping. The incoming traffic is allowed to pass through only when it matches a valid mapping, which is set by an outgoing and *"already-authorized"* data stream. As illustrated in Figure 6(a), a valid entry is created for each outbound traffic flow. It then acts as a traffic filter for subsequent inbound flows. Second, the standard specification recommends to install traffic filters at border gateways and access routers [5], to prevent unauthorized traffic from traversing the cellular network. Last, user-installed filter rules (inserted into the EPS bearer) at the core gateway can also facilitate to shield unwanted traffic from reaching the UE. Note that such rules are proposed to differentiate packets with diverse quality-of-service requirements. Each inbound packet is thus aligned with a valid EPS bearer for the target UE. In a nutshell, only packets passing all filter rules are delivered to the UE.



U
P-GW
Data-plane
Control-plane
U.GTP
U.IP
☑ UL authorization via authentication
Filtering: setup a valid entry
☑ Filtering:
map to a valid GTP
☑ Filtering:
match with a valid entry
U
P-GW
Filtering: setup a valid entry
UE: Close the app
(TCP: half-open, UDP: still open)
Entry (stateful)
Expiration independent of UE
☑
☑
SPAM

(a) Current authorization solution

(b) Authorization fraud: "non-expiring" authorization

Figure 6: Current authorization solution and the authorization frauds on the control plane.

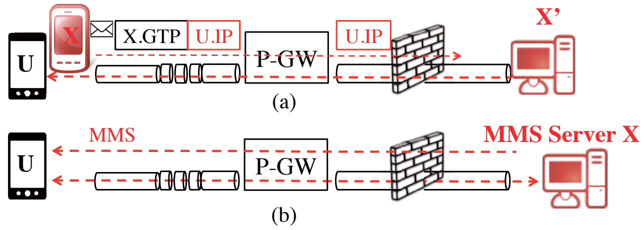


Figure 7: Illustration of two Cloak-and-Dagger Spamming Attacks. Attacker X exploits the authorization frauds via IP spoofing and MMS.

3.2.2 Vulnerability Analysis

We uncover two possible cases of authorization frauds for data transfer initiated by an external host on the Internet. Effectively, no proper authorization is in place for the inbound transfer. In the first case, the adversary deceives the NAT to open a backdoor through the authentication bypass loophole. The incoming traffic thus passes through the firewall and NAT. It circumvents the current fence that leverages authenticated outgoing traffic to indirectly authorize the incoming traffic. Note that, NAT, as well as the firewall and border router, can exploit the information only at IP and above layers; It is impossible to validate whether outgoing packets originate from the authenticated one.

The second fraud occurs when any side-channel or third-party mechanism is exploited to trap the UE to leak its data access information. Afterwards, the user is tricked (usually unaware) to initialize certain outgoing data delivery, thus granting access to the attacker. Moreover, popular mobile applications (*e.g.*, VoIP, MMS) may use the push-based communication model and perform automatic background operations. This feature can also be exploited to trap the victim.

The above frauds exploit the "non-expiring authorization," as illustrated in Figure 6(b). The current practice is to invoke one-time authorization only at the start, but apply soft-state renewal by data packets during actual transfer. For example, when an inbound packet arrives at NAT, the mapping between its IP address and the port number remains valid until timeout (*e.g.*, 5 minutes). However, once the access is granted, it is largely beyond user control.

IP delivery follows the store-and-forward model, and intermediate routers relay packets asynchronously. When the UE tears down the flow, NAT may not flush the mapping entry right way. Moreover, the access control decision is made locally. If new packets from the flow arrive before timeout, the timer is refreshed and the flow is still considered alive. Consequently, incoming packets can still pass through and charges are imposed accordingly. This happens although the user has terminated the flow on his side. The threat becomes ominous when the user imprudently authorizes malicious data access. As a result, *data transfer is allowed without the endorsement from the user*. While doing nothing, mobile users suffer from spam attacks and associated billing charge. Even worse, users lack effective mechanisms to stop the spam.

The root causes are multifaceted. First, one-time authorization during initiation cannot ensure access control for long-lived transfer. Runtime authorization is needed. Second, current authorization is mostly open loop without taking input from the end user. Without the user's decision feedback, authorization cannot be done properly. Third, no de-authorization mechanism is available, so the user does not have the mechanism to stop spam at will. The push-based delivery in IP makes attacks easier.

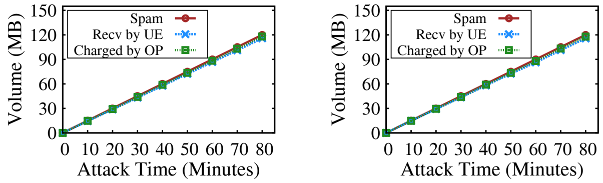
Note that the charging model asks the phone to pay for both upstream and downstream traffic. For the authorization vulnerability, we assume that the mobile phone is secure (without malware). Therefore, the upstream traffic comes from the mobile device (except the IP spoofing packets), and authorization for upstream packets is correct. We thus focus on downstream data sessions in the showcase attack.

3.2.3 Exploit: Cloak-and-Dagger Spamming Attack

The attack idea is to inject spam messages to mobile victims, thus increasing their data bills. The key is to deceive cellular networks to allow for spamming. We propose two approaches which correspond to each case of authorization frauds identified above, as shown in Figure 7. The first approach is to counterfeit an outgoing data packet from the victim via IP spoofing. The inside attacker (the attack phone) impersonates the victim to set up a connection with the external spamming server. Data spam follows thereafter.

The second approach is to set a trap to obtain data access to the victim. We sketch a new spamming attack via Multimedia Messaging Service (MMS). MMS is a standard service offered by cellular carriers. It is used to send multimedia content to mobile phones. The attack exploits the automatic data connection setup in MMS. When the phone receives a MMS message, it automatically opens a HTTP connection with the MMS server and retrieves data. Therefore, the attacker pushes one MMS message to the phone, which embeds the link to his own malicious server. Once the phone connects to the server, the attacker starts to spam over this connection. Here, we exploit the push model in MMS, which was used to drain phone's battery in [31]. Different from their work, we exploit it for the charging attack. Moreover, we found that the approach proposed in [31] failed in our case. It was because we use TCP but not UDP. To make it succeed, we refine the attack in three aspects. First, we set the transfer encoding of the HTTP connection as chunk based. Without this configuration, the connection may be disrupted by the phone. Second, small chunks are sent to the phone to keep this connection alive. Last, spamming packets are modified to prevent from triggering any abnormal HTTP event. Sequence numbers outside the congestion window are used in TCP packets. As a result, these packets are received by the phone and are charged, but they are discarded by TCP and do not affect the HTTP connection.

Note that the trap is not only limited to the above forms. It can be done through phishing, abusing VoIP tools (*e.g.*, Skype) [27] or an existing TCP transmission [22]. However, they all require some actions (*e.g.*, clicking a phishing link or starting a connection first) from the victim, thus restricting its practicality. Instead, our cloak-and-dagger spamming attacks are more subtle and severe. They do not require malware or actions on the victim phone. Moreover, the MMS-based attack is quite threatening since it launches the attack against a chosen victim and only requires the phone number of the victim (which is easily accessible).



(a) OP-I, via IP spoofing

(b) OP-II, via MMS

Figure 8: Data volume incurred by spamming attack varies with duration in OP-I and OP-II.

3.2.4 Experimental Validation

Our experiments validate that the authorization loophole indeed exists. Mobile users are vulnerable to spamming. The spam incurs billing upon mobile victims who do not authorize such data transfer.

Loophole verification. We observe the IP-spoofing based fraud in both carriers, and the MMS-based one only in OP-II. For the first one, we extend the above two-phone experiment. In addition to forging `U`'s IP address, adversary `X` also asks the server to send junk data to user `U`. For the second case, we deploy a malicious MMS server in our lab and trap the victim to download the multimedia content. Once the connection is established, data delivery can be controlled by our server. Both experiments show that, `U` receives all these data and is charged for the spam, despite taking *no* action. Note that even if the victim quickly switches on after off, s(he) is still vulnerable because possibly the same IP address is assigned and NAT/firewall keeps the attacked port valid.

Attack assessment. We have prototyped the above attacks and show the spamming result via IP spoofing for OP-I, and via MMS for OP-II. The assessment for other attacks is similar. Figure 8 shows the attack damage during 80 minutes; the spamming packets are sent at 200 kbps with 500 bytes each. There is no sign of limit on the attack duration. The victim receives about 115-117 MB junk data, thus being charged for about 118 MB by both carriers. The junk data is discarded before being passed to the application layer.

3.3 On Accounting

3.3.1 Current Solution and Vulnerability Analysis

The current solution to ensure volume accuracy depends on the accounting operation in parallel with data transfer (see Figure 9). The logging is done at the core gateway since all packets must traverse it to reach their destination (either the UE or the external host). The volume sums up the payload (including IP and above headers) of all arriving packets.

However, the accounting volume might differ from (usually larger than) that actually delivered by cellular networks. The volume is inflated once data delivery fails after being counted at the gateway (see the bottom plot of Figure 9). The inaccuracy can occur when some get lost or dropped under certain attacks or failures over the radio link. It can be manipulated by exploiting connec-

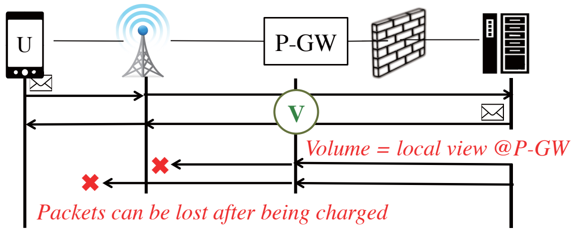

U
P-GW
V
Volume = local view @P-GW
Packets can be lost after being charged

Figure 9: Current open-loop accounting solution.

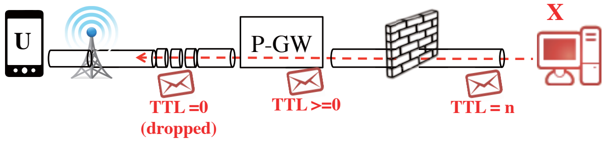


Figure 10: Illustration of Hit-but-no-touch attack. Attacker X modifies the TTL value so that packets are dropped after being charged, thereby imposing over-billing.

tionless IP-based delivery, where data packets are independently forwarded or dropped by each network element on the path (see the proposed attack).

The above vulnerability is rooted in the open-loop charging model. Billing is based on the *local view* at the gateway only, and accounting inaccuracy is inevitable whenever packet delivery differs before and after the gateway.

3.3.2 Exploit: Hit-but-No-Touch Attack

We devise a novel hit-but-no-touch attack to overcharge the user. The idea is to modify the time-to-live (TTL) field of incoming IP packets so that they only reach the core gateway but not the phone. Note that TTL is decremented by each intermediate router and discarded if it reaches zero. Our attack leverages this delivery rule for IP packets, originally designated to prevent packets from being routed over loops over the Internet. Given an improper TTL value, packets arrive at the gateway (accounting completes) and then are discarded , thus incurring over-billing. The attacker first probes with different TTL values, determines the appropriate parameter, and then activates the attack. This attack design allows for the adversary to send packets in a covert manner. The victim is charged for data that never arrive at the phone.

This attack differs from our prior findings on inaccurate accounting volume [28] [36]. Over-accounting has been reported in two scenarios: lost wireless connectivity [28] and mobility-triggered handoff [36]. However, both are caused by plausible, non-human factors which exhibit only under certain settings (*e.g.*, lossy wireless channels and mobility). In contrast, the hit-but-no-touch attack can be launched anytime, anywhere. It can work with the spamming attack (§3.2) or any other ongoing data services, thus forming a stealthy accounting attack. Moreover, since the IP packet delivery is connectionless, it is thus hard, if not impossible, for the gateway alone to differentiate whether the zero-valued TTL is malicious (incurred by improper initial value set by attackers) or not (caused by delivery over too many hops).

3.3.3 Experimental Validation

We confirm that this hit-and-no-touch attack is feasible. This also verifies the accounting loophole that is based on the local view at the core gateway. We vary the TTL values of the spam packets sent by the adversary, and observe the volume gap between the gateway and the phone. The spamming volume is 5MB. We run experiments for both carriers at ten locations. Figure 11 shows the result at one location; the results at other locations are similar. It validates its feasibility for both operators. There are three feasible TTL choices for OP-I: 26, 27, and 28, and only one feasible TTL value for OP-II: 18. Once the right TTL parameter is used, the phone suffers from unexpected, even unknown overcharging. Since same results are observed at all locations for each carrier, the feasible TTL value can be reused by other conspirators.

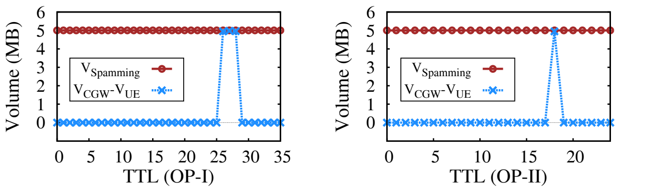


Figure 11: Accounting volume gaps under varying TTL values in OP-I (left), OP-II (right).

3.4 Root Causes

We now reexamine the root causes to MDC vulerabilities in order to learn the fundamental limits and gain the solution insights. It turns out that, both the cellular networks and the Internet design fundamentals may have to share the blame.

On the cellular side, two design guidelines for MDC systems may accidentally make possible the insecurity loopholes. First, MDC performs accounting operations based on the local view at the core gateway. This effectively produces an *open-loop* charging solution. Without feedback from mobile users, it is difficult to conduct proper authorization for the billed traffic. It is also challenging to enforce accurate control for the recorded traffic volume. Second, while cellular networks adopt multi-layer solutions to security fences, they do not stipulate cross-layer security binding. There is no mandate on runtime binding for security functions. This practice opens loopholes in authentication and authorization. When digging even deeper, we find out that, current MDC design is largely taken from the legacy 2G cellular network. Note that 2G uses circuit switching for voice calls. In contrast, 3G/4G has migrated to offer data service using packet switching. The open-loop charging design works well for circuit-switched 2G networks, since the user has to establish virtual circuits (VCs) before calls. VCs inherently offer the closed-loop feedback between users and the network. However, closed-loop feedback no longer exists in IP-based connectionless data delivery over 3G/4G networks.

On the Internet side, two features of network-layer IP data delivery contribute to the vulnerability. One is that IP uses the *push* delivery model at the network layer. Using IP, any device on the Internet can initiate packet delivery and push the data to a chosen target without prior consent. This helps in authorization and spamming threats. The other is that intermediate IP routers are not required or unable to verify the authenticity of the source IP address. IP address spoofing is possible during data communications between a mobile device and an Internet host.

In a nutshell, the MDC security problems are rooted in the inappropriate charging architecture (which is good for CS voice transmission) is used for PS data transmission.

4. DEFENSE SOLUTIONS

In this section, we propose defense measures to protect the MDC system. Our solution also seeks to be 3GPP standard compatible, thus facilitating fast deployment. Figure 12 shows the overall solution framework, which has three main components:

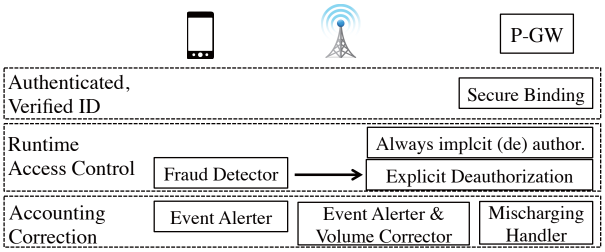


Figure 12: Secure charging solution framework.

• **Authenticated and Verified ID.** *Secure binding* is enforced at the core gateway. The charging ID for each UE is authenticated and verified cross-layer for each packet.

• **Coordinated Runtime Access Control.** It allows for the UE to collaborate on access control. Implicit authorization is always applied on the data plane, while explicit de-authorization is triggered *on demand* over the control plane. It thus strikes balance between minimizing control overhead and shielding from attacks.

• **Accounting Alerts and Error Correction.** It seeks to minimize charge errors (mainly overcharging) based on the feedback from both end-device and network components. It is both proactive when preventing over-billing through the alerting function and reactive when offsetting extra charges via the correction function.

We apply three guidelines in our solution. The first is *coordinated design between data and control planes.* In authentication, the control plane authenticates the charging ID, whereas the data plane verifies it for each packet delivery. In authorization, the data plane offers implicit authorization for data flows, while the control plane activates explicit de-authorization for spam whenever detected. This way, concerted actions are taken on both planes. The second is to introduce *infrastructure-assisted feedback*. The feedback facilitates users and the infrastructure to share consistent view on charging. It also empowers users to detect and defend against malicious attacks. The third is to *enforce cross-layer security binding*. The data delivery spans layers across multiple components. To harness existing security mechanisms for MDC, we enforce secure cross-layer bindings at runtime.

Note that other alternatives are also possible. For example, the operator may apply deterrence by detecting and punishing them. Compared with them, ours takes the collaborative approach between the infrastructure carrier and the cellular user. It leverages the increasing capability of smart end devices. Moreover, the user knows best regarding whether a message is spam or not. The infrastructure alone may not be able to handle all threats. Furthermore, punishing attackers *a posteriori* is fine with users with monthly data plans, but may not work well with the pay-per-use model.

4.1 ID Authentication and Verification

In MDC, two IDs, *i.e.*, the Tunnel ID and the IP address, operating at the GTP-U layer and the network layer, respectively, may serve as the charging ID. The Tunnel ID is both authenticated on the control plane and verified for each packet on the data plane. This is because it is bound to the physical link by the infrastructure. In contrast, the IP address may not be verified for every packet.

Our solution is to apply cross-layer binding by tying the network-layer IP with the lower-layer bearer information (*i.e.*, Tunnel ID). The secure binding is kept at the gateway. Consequently, secure binding between the packet's IP address and the UE's authenticated IP is assured. With this binding in place, upon each packet arrival, the gateway checks whether its carried source IP is identical to the IP stored in the UE's EPS bearer. If different, the UE is considered unauthentic and charged while the packet is discarded.

An alternative approach is to perform MDC directly based on the trustworthy Tunnel ID. However, this solution undermines the flexibility offered by the current scheme. For example, it restricts the charging function only at cellular domain gateways, not any IP-enabled components (*e.g.*, border gateways). Moreover, it cannot well support existing IP-based or flow-based charging.

Implementation using 3G/4G mechanisms. The above design can be implemented using the rules offered by the PCC (Policy Control and Charging) mechanism [6]. The PCC rules define a set of data flow filters and taken actions. Originally, they serve as packet filters and impose diversified charging policies. They are defined during the initial bearer establishment and kept in the bearer at the gateway. We leverage these rules to impose minimal change to the gateway.

To ensure secure binding, we define PCC rules with two action types: pass and drop. Each pass-type rule must have the authenticated IP, not a wildcard ($*$), specified in the source IP field. A drop-type rule is defined to filter out all the traffic not from the authenticated source. This is done via adding a PCC rule with (srcIP= $*$) into the EPS bearer, as the lowest-priority rule. For example, the simplest case has the following two rules: Rule 1 (srcIP, *,*,*,* : `PASS`); Rule 2 (*,*,*,*,* : `DROP`). As a result, the spoofed packets will be mapped to the drop-type rule without matching all preceding, high-priority pass-type rules. The spoofed packets will be discarded. To prevent network resource from being wasted, their authentic senders can be penalized by being billed of those discarded packets. We note that Cisco provides a source IP address verifier in [15]. It uses an additional module, but not the standard-compliant mechanism.

We implement ID authentication and verification at the gateways, instead of base stations, because the essential information is only available at gateways but not at base stations. The legitimate IP address for each device is maintained in its EPS bearer, which is only stored at gateways. Moreover, we leverage existing charging operations to lower the processing overhead. Note that the mapping from the source IP address to the user is performed in the current charging procedures. On top of this mapping, our solution merely adds a comparison on whether the source IP address is identical to that in the EPS bearer. The operation of comparison is of low cost compared with that of creating another mapping. Our evaluation also confirms that our scheme incurs little overhead (§5).

4.2 Coordinated Runtime Access Control

The current practice suffers from authorization frauds when wrong access is initially granted or the access becomes malicious afterwards. Without sufficient information, the network infrastructure is often unable to react to these frauds correctly or timely, thus incurring improper charges. To fix it, we utilize user feedback to help the infrastructure to determine whether the access should be granted or denied in time. It not only assures the users to pay for what they want, but also respect and protect their rights to not pay for what they do not want. A flow is the basic granularity for access control. We need to address two issues: (1) how to to (de)authorize a flow? (2) how to detect a malicious flow and interact with the infrastructure? The solution should also be efficient and scalable.

Coordinated (De)Authorization. Access control is coordinated between data and control planes. The data plane uses implicit authorization at runtime, thus incurring low overhead and being more scalable. Explicit de-authorization is invoked on demand on the control plane, in order to block certain unwanted flow.

Our data-plane access control still uses the current practice. A flow is authorized once any packet from it is sent *uplink* by the UE. It is considered terminated and thus de-authorized upon timeout, when no packets are seen before the timer expires. This soft-state based de-authorization incurs no signaling overhead. Note that, however, adversaries may still inject spam messages before timeout, and thus this soft-state de-authorization is not bullet proof against attacks.

We further propose an *on-demand* triggered, explicit de-authorization scheme on the control plane. It relies on the signaling packets of de-authorization requests sent by the UE. They can be activated on demand when the UE wants to terminate a flow. This explicit de-authorization scheme allows for the user to conduct access control at will. It is deemed most effective when the UE detects spam and immediately de-authorizes the attacker. However, this explicit approach does incur extra signaling overhead. It should not be taken in common usage scenarios without attacks.

Note that attackers cannot purposely drop the de-authorization signaling messages in cellular networks, since all data transmissions are regulated by the base station and smartphones cannot control the radio channel. Moreover, our explicit de-authorization does require a supporting component of fraud detector at the UE. It detects two types of malicious traffic. One is an unauthorized flow, which matches no corresponding transport-layer flow at the UE. When a spam packet arrives at the UE, a specific ICMP message of "`PORT UNREACHABLE`" is generated. The detection is thus simple by monitoring whether any such message is generated. The other is the spam that abuses an "authorized" flow. In this case, only transport or application protocols can tell whether the arrived packets are indeed spam or not. For example, the transport layer discards invalid TCP packets (*e.g.*, in the usage-inflation attack [22] or our proposed MMS-based threat). The detection can be based on packet drops. For instance, the flow is identified as a spam when its discarded traffic volume exceeds certain threshold over a chosen time window. In case of small threshold, false positive may occur though it is rare. We thus introduce another mechanism by calling for user awareness. Upon detecting suspicious spam, an alert message pops up and waits for user response. Automatic blocking is activated when the threshold is larger than a certain value. Note that spamming is fundamentally determined by the IP push delivery model, and no approach can completely eliminate the spamming. The threshold greatly limits the spamming scale. If the attacker changes its IP address or port number, it is blocked by NAT. A new attack has to be re-launched.

This detection works together with alarms and user feedback to prevent false positive conditions; It reports suspicious traffic and calls for user decision (block, by default). Afterwards, the UE communicates the control-plane access control module to de-authorize the corresponding flow. For flexibility, the detection can be done via third-party detectors with appropriate permission control.

Implementation using 3G/4G mechanism. The above solution can also be implemented by leveraging the PCC mechanism. We use the dynamic PCC rules and add three event triggers for implicit authorization, implicit de-authorization, and explicit de-authorization, respectively. Upon the arrival of an unauthorized uplink packet at the gateway, one pass-type rule is added to automatically authorize the flow. The second event is triggered by the inactivity timeout of a flow filter. When triggered, the filter is deleted. For explicit de-authorization, we reuse an existing event trigger, `Resource Modification Request`, to let the UE communicate with the PCC module. The UE specifies which flow to be de-authorized in the request, and then this flow filter is deleted by PCC. The number of PCC rules is limited by 255 in the standard, and this should be sufficient for common usage. In case when this limit is reached, rules can be merged or the new flow is denied.

4.3 Feedback-Based Mischarge Correction

To avoid charge errors, we enable runtime feedback to minimize accounting inaccuracy caused by the open-loop MDC design. We first focus on overcharge, but come back to under-billing later. Three problems are addressed. (1) When to issue a feedback? (2) How to use this feedback? (3) How to avoid new threats while using the feedback from the user?

Runtime feedback is realized through two phases of accounting alert and correction. The alert phase is to determine whether any suspicious accounting behavior occurs and determines when to issue a feedback. An alert is triggered when certain event occurs. These events cover two categories: (1) direct packet drops/losses; For example, the base station or the gateway observe unsent, unacknowledged, or discarded packets for each served UE (*e.g.*, in the TTL-based attack); and (2) events that might incur packet losses (*e.g.*, handoff or lost radio connectivity [28]). At the correction phrase, two mechanisms, mischarge prevention and compensation, are introduced to handle those two types of events, respectively. If direct packet losses are observed, the alert with such traffic volume is sent to the mischarge handler at the gateway (*e.g.*, P-GW), so that the reported amount is refunded. If suspicious events are detected, the alert message is sent to the handler. The handler then freezes all ongoing flows for the UE. They are resumed after the detected scenario disappears. Such designs seek to minimize accounting errors in MDC while keeping the feedback overhead in check.

Our solution further prevents user cheating behaviors. The correction function is adopted only in the trustworthy network infrastructure while the alert function is enabled on both the UE and network components. The UE is only allowed to report abnormal conditions, but not to submit the refund volume. Therefore, he has no incentive to cheat, since the only action that he can trigger is to suspend data flows (via explicit on-demand de-authorization). In fact, in case of real overcharges, it can be detected and verified by the network infrastructure. Note that attackers cannot misuse this mechanism by reporting that they have not received the packets. The base station knows exactly what packets have been successfully delivered to a mobile phone via its Layer-2 acknowledgments.

Undercharging correction. The undercharging can also be detected and avoided by the accounting alerter and corrector respectively. It favors mobile users so that the users usually have no incentive to help out. User feedback becomes of little value. The alerting and correction have to reside at the infrastructure side. Assume that volume counting does not go wrong. Undercharging is usually caused by the practice of diversified charging policy, for example, free rides via free DNS services [28] or free TCP retransmission [22]. The defense key is to verify whether the flow is a genuine data service that is eligible for some specific policy. To minimize the overhead, alerting is used to filter out suspicious traffic. For example, an alert is triggered when the free volume reaches a threshold; the normal traffic portion for DNS and TCP retransmission is usually very small (<2.5% [19, 22]). The threshold can be set accordingly. For suspicious flows, deep packet inspection (DPI) can be further used (certainly, the overhead is big). Once the abuse is detected, certain action will be taken, such as normal billing, flow de-authorization, SMS warning, blacklist and *etc.*. The undercharging may not be limited to only these two types of traffic due to charging policy diversity. Technically, others can still be prevented by our secure framework.

Implementation using 3G/4G mechanisms. We reuse or implement an additional module for the alerting and correction at the infrastructure side. For example, the base station can reuse the "Unsent Data Volume" field [4] to send the feedback to the gateway. The gateway may need to add one function to record the dropped packet volume if it is not available. The UE adds event callback functions to detect suspicious events (*e.g.*, handoff and insufficient coverage). To detect handoffs, the UE exploits the handoff request or complete messages received on the RRC (radio resource control) layer. To freeze and resume ongoing traffic, we still reuse Resource Modification Request to communicate with the gateway.

Note that our solution may require per-flow state, but the overhead is not big. At most two states for a flow are kept at the gateway. A base station does not need to add new per-flow states. It is to record the volume not sent out per phone. To support user feedback, it adds one more state per phone to identify whether the user reports abnormality. The gateway needs to add two states of status and volume per flow. It records whether wrong volume is used and how much volume should be offset. Our evaluation shows that, the gateway can process fast enough without incurring much delay, compared with the current processing of each packet.

4.4 Defense Incentive

As potential victims, mobile users always have incentives to deploy at least the local defense measures (*e.g.*, local detector, event alerter) to protect themselves. On the operator side, it might be true that operators have no immediate incentive to fix overcharging attacks (*e.g.*, spamming, hit-but-no-touch attacks). However, they are held responsible for fixing the threats since these attacks do exploit the MDC loopholes in the cellular infrastructure. Users did nothing wrong (within their capability). Under the pressure of public disclosure, user complaints and even possible lawsuits, we believe that operators would deploy defense measures to serve as responsible carriers.

5. PROTOTYPE AND EVALUATION

We now describe the prototype of our solution, and its evaluation in a variety of malicious and normal usage scenarios. The results partially confirm its effectiveness and low overhead.

Prototype. Figure 13 shows our secure MDC prototype. Without access to the operator's gateways, we deploy a proxy outside the cellular network to emulate the core gateway. All traffic flows from the device to the Internet go through this proxy. We implement all proposed secure components for the UE at the phone, and other components in the cellular infrastructure (except secure binding) at the proxy. Without access to Layer-2 information in cellular networks, we assume that secure binding is already in place. All the event-triggered functions are implemented by callback functions. The proxy uses a Dell Inspiron 660 machine, which runs Ubuntu 12.04 on a Intel Core i3 CPU at 3.4GHz and with 4GB memory. An Android smartphone serves as the UE in our experiments. Specificially, we develop two modules of fraud detector and event alerter at the phone. At the proxy, we develop the following components: (1) charging function, (2) coordinated runtime access control, (3) event alerter and volume corrector, and (4) mischarging handler (mainly for overcharging).

Evaluation summary. We assess our solution in various scenarios of malicious and normal usages, including three proposed attacks, two attacks in the literature, as well as an overcharging scenario and two normal settings without attacks. The evaluation validates the effectiveness of our solution. The system is able to identify all the spamming attacks and the overcharging occurrences, containing the charging error within 35 KB (depending on the attack and defense parameters; more details are given later). We also examine the extra overhead on runtime access control, local detector, alerter and corrector. Note that most mechanisms are event driven (*i.e.*, on-demand triggered). The results show that our solution is light-weight and the overhead is usually negligible.

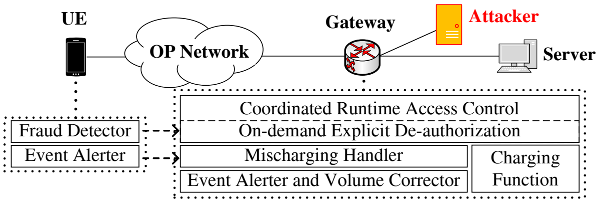

UE
OP Network
Gateway
Attacker
Server
Fraud Detector
Event Alerter
Coordinated Runtime Access Control
On-demand Explicit De-authorization
Mischarging Handler
Event Alerter and Volume Corrector
Charging Function

Figure 13: Secure MDC Prototype.

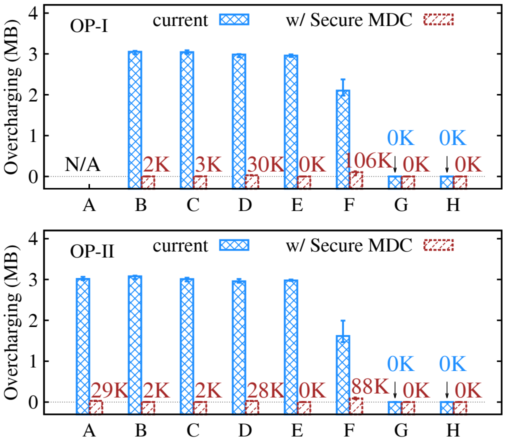


Figure 14: The (min/med/max) overcharging volume with/without the secure MDC solution in two carrier networks: OP-I (top) and OP-II (bottom).

5.1 Defense Effectiveness

We test our prototype in various scenarios. They include four spamming attacks (two in the literature), two over-accounting cases and two normal settings without attacks. Specifically, they are (A) spamming via MMS, (B) spamming via IP spoofing,(C) spamming via Skype [27], (D) spamming via TCP retransmissions (*i.e.*, usage-inflation attack [22]), (E) Hit-but-no-touch attack, (F) Over-accounting in insufficient coverage [28], (G) light-traffic setting (Web browsing), and (H) heavy-traffic setting (Youtube). In (A)-(F), each test lasts 2 minutes, and we generate 200Kbps downlink data to the phone. The total traffic volume is 3 MB. For the above TCP-based spamming threats (A and D), we generate TCP spam packets via Raw Socket programming. In E, the proxy first performs charging and then discards those packets with their TTL being zero. In F, the UE is carried from one location with good radio signal to another without signal, and the duration in the no-signal zone is 1 minute. For comparison purposes, we test two normal usage scenarios: reading CNN homepage (∼39 KB) in G and watching a Youtube video (∼3 MB) in H, both at spots with excellent coverage. We do not run the free-uplink attack, but assume it is fixed once secure binding is enforced.

We configure our defense solution as follows. For the fraud detector at the UE, spamming is inferred under any of the following conditions: (1) when one ICMP PORT UNREACHABLE packet ($\theta_{ICMP} = 1$) is observed, or (2) when the transport-layer packet dropping exceeds $\gamma_{drop} = 10\%$ or $\theta_{drop} = 100$kbps per flow within the detection time window (say, 1 second), or (3) when accumulative packet dropping exceeds $V_{drop} = 500$ KB per flow. If either of Conditions (1) and (2) is met, de-authorization is automatically activated. However, to prevent false positives, the device asks for user feedback when taking actions if Condition (3) is met. Since we are unable to access the base station, we emulate its event alert at the gateway. It detects the UE's status by monitoring the keep-alive packets periodically generated by the UE, instead of the Layer-2 ACK packets. When no keep-alive packet is observed for 10 seconds, the alert for insufficient coverage is triggered.

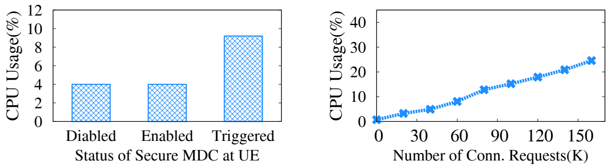
Figure 14 plots the charging volumes with/without our defense in all scenarios. It confirms that our mechanism successfully defends against malicious abuse and overcharge. For spamming via IP spoofing (B) and Skype (C), the overcharge volume is always below 3 KB in both carriers. This is because the UE immediately activates its de-authorization to the gateway when the first spam packet arrives (*i.e.*, an ICMP message is created). The charging errors are only affected by the round-trip time between the UE and the gateway (mostly below 0.1second in Figure 17). For spamming via MMS (A) and TCP retransmissions (D), the overcharge errors are a little larger but still below 30KB. This is because detection delay is affected by the detection time window (here, 1sec). In the TTL-based attack (E), the overcharging volume is 0, since the volume corrector requests the handler to refund the dropped volume. For insufficient coverage (F), the medium values of overcharging volume are 106 KB and 88 KB for two operators, respectively. In fact, such overcharging could be avoided, if the volume corrector were deployed at the base station, which reports unsent and unacknowledged volume. This is not available in our prototype.

Given the set of parameters, we are unable to test all attack cases. These defense parameters may not be appropriate in all settings. The detection parameters serve as the tuning knob to balance between false-positive and false-negative errors, as well as detection delay. For example, the smaller the spam detection threshold, the more likely the false-positive error; The larger the errors, the longer the time for detection, the more likely the false-negative error. These parameters can be configured as user-specific profiles or application-adaptive patterns. Our evaluation focuses on validating the basic mechanism, while leaving fine tuning of parameters as the future work. Note that, however, no noticeable difference is observed in case of continuous spamming and large overcharging errors. Once it imposes nonnegligible damage to the UE, it is detected. For example, when the adversary reduces the spamming speed (*e.g.*, <10kbps), it just prolongs the detection duration under the current defense setting. In this case, our defense still limits the overcharging volume. We also run large numbers of tests for common usage. We rarely observe `ICMP PORT UNREACHABLE` packets and never have relatively large fraction (*e.g.* 10%) of packets discarded by the transport layer. Therefore, the false-positive error hardly occurs. Even in such rare cases, it may still resort to user decision.

5.2 Defense Overhead and Impact Factors

Our solution incurs both message and processing overhead. The extra messages come from the fraud detector, the event alert and the volume corrector, all of which are triggered on-demand. The message overhead is thus in proportion to the number of abnormal (suspicious) flows. In the above tests, it never incurs extra messages in normal usage and introduces 1-2 extra messages per flow under malicious abuse or abnormal usage.

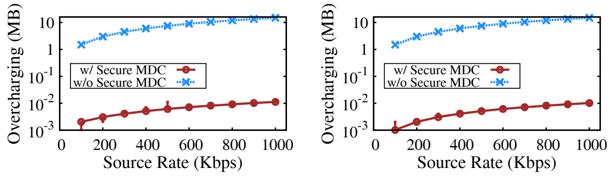
The processing overhead stems from those lightweight monitor functions and event-triggered processing components. Under normal traffic, only those monitor functions run. Figure 15(a) compares the CPU usage at the UE when our defense module is at three states (*i.e.*, disabled, enabled, and triggered). Each CPU usage slice is 200ms. In each test, we collect 5-minute trace for the first two states and all those samples when our extra processing is triggered. It shows that, the UE consumes about 4% CPU usage no matter whether our defense is enabled or disabled. It implies that extra processing overhead in normal cases is negligible. Upon event detection, the CPU usage climbs to 9.2% (average) but only occupies a single slice. This CPU usage is to trigger a process that issues a request or feedback to the gateway. We observe similar CPU usage at the gateway. Therefore, the overhead by our defense module is low and affordable. We further run scaling tests at the gateway. Figure 15(b) plots the average CPU usage with respect to the number of connection requests when our components are enabled. It shows that, the CPU usage increases linearly with the number of active connections. Our solution thus works fine with edge routers in the network.



(a) UE (b) Gateway

Figure 15: The average CPU usage of Secure MDC at the UE (a) and the gateway (b).

We also assess the impact of the spamming rate. Figure 16 shows that, our defense can effectively stop spamming given the increasing spamming rate. The overcharge volume increases slightly due to communication delay to the gateway. Figure 17 measures this latency by sending an explicit request to the gateway using our prototype. It is about 100ms in our tests.



(a) OP-I (b) OP-II

Figure 16: The overcharging volumes varies with spamming rates in OP-I (a) and OP-II (b).

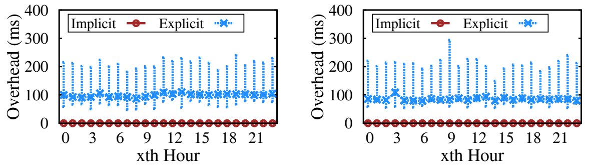


Figure 17: The (min/med/max) delay of an explicit request at 24-hours in OP-I (left) and OP-II (right).

6. RELATED WORK

Several studies recently assess the MDC system in cellular networks [21,22,27,28,36]. They mainly explore its accounting misbehaviors in normal use or under attacks. Specifically, [28, 36] study the accounting inaccuracy in the presence of weak or no wireless connectivity, user mobility, and policy misconfigurations by operators. The others exploit the vulnerability that the MDC system does not carefully consider transport-layer's traffic (TCP retransmission [21,22]) and application-layer's (DNS, VoIP and phishing links [27]). These studies attempting to solve individual exploits have been ad-hoc to secure the MDC system. Different from them, we examine vulnerabilities in the MDC system along all three AAA dimensions. We further stretch several new and practical attacks, and then provide the solutions. In particular, we have not seen similar solutions to authentication and authorization in the literature. Others (*e.g.*, [23, 32]) work on the pricing scheme, which is related to MDC but orthogonal to the AAA issues studied in this paper.

Except the MDC system, research on the performance and security aspects of cellular networks has been an active area in recent years (see [1, 9, 18, 24, 26, 29–31, 33–35, 38] for a few examples). They conduct security studies in three broad aspects. First, they identify the vulnerabilities in cellular-specific technologies, such as ID leakage via SIM/USIM card hijacking [1], AKA loopholes in user authentication [9, 26], battery drain via MMS pushing [31]. Second, they assess the Internet technology in the cellular network context. For example, [38] studies the NAT and firewall policies over cellular carriers, and introduces the TCP hijacking attacks in [29, 30]. [34] analyzes the impact of cellular botnets. [24] analyzes malicious traffic from one large operational carrier. The last category explores the interaction between the cellular technologies and data services. For example, [18, 33] exploit SMS to launch denial of service (DoS) attacks via overloading the control channels for data services, and [35] generalizes to a series of DoS attacks by exploiting difference between cellular networks and the traditional Internet. Security and privacy of mobile devices and applications is another active research area (such as [11, 13, 17, 37, 39]). They are independent of the MDC system in this work.

7. CONCLUSION

In this work, we conduct systematic security analysis of the MDC system in cellular networks. We uncover vulnerabilities in every subsystem of authentication, authorization, and accounting. We show that, no sophisticated attacks are needed, and simple attacks may work in practice. As far as we know, many cellular operators are still unaware of such security weakness. For the same reason, simple yet effective defense measures have not been deployed by many of them. These results catch us off guard to certain extent. Although cellular data charging has been operational for years and is generally successful, our study illustrates how fragile the networked system can be from the security standpoint.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work is supported in part by the National Science Foundation under Grants No. CNS-1421933 and CNS-1422835. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

[1] Millions of cell phones could be vulnerable to this sim card hack. http://gizmodo.com/sim-cards-are-hackable-and-researchers-have-found-the-v-860779912,2013.

[2] Tor. https://www.torproject.org/.

[3] 3GPP. TS32.240: Charging architecture and principles, 2006.

[4] 3GPP. TS25.413: UTRAN Iu interface RANAP Signaling, 2008.

[5] 3GPP. TS33.210: 3G security; Network Domain Security (NDS); IP network layer security, Dec. 2012.

[6] 3GPP. TS 23.203: Policy and Charging Control Architecture, 2013.

[7] 3GPP. TS33.401: 3GPP SAE; Security architecture, Sep. 2013.

[8] Allot. Allot mobiletrends charging report, 2011. http://www.allot.com/MobileTrendsChargingReport.html.

[9] M. Arapinis, L. Mancini, E. Ritter, M. Ryan, N. Golde, K. Redon, and R. Borgaonkar. New Privacy Issues in Mobile Telephony: Fix and Verification. In *ACM CCS*, 2012.

[10] AT&T. Data Plans from AT&T. http://www.att.com/media/att/planner/index.html.

[11] K. W. Y. Au, Y. F. Zhou, Z. Huang, and D. Lie. Pscout: analyzing the android permission specification. In *ACM CCS*, 2012.

[12] H. Balakrishnan, S. Seshan, and R. H. Katz. Improving reliable transport and handoff performance in cellular wireless networks. *Wireless Networks*, 1(4):469–481, 1995.

[13] M. Becher, F. C. Freiling, J. Hoffmann, T. Holz, S. Uellenbeck, and C. Wolf. Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of Mobile Devices. In *IEEE S&P*, 2011.

[14] S. M. Bellovin. Security problems in the TCP/IP protocol suite. *SIGCOMM Comput. Commun. Rev.*, 19(2):32–48, 1989.

[15] Cisco. Cisco GGSN Release 10.0 Configuration Guide, 2010.

[16] Cisco Visual Networking Index. Global Mobile Data Traffic Forecast Update, 2013–2018, 2014. `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html`.

[17] W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri. A study of android application security. In *USENIX Security*, 2011.

[18] W. Enck, P. Traynor, P. McDaniel, and T. La Porta. Exploiting Open Functionality in SMS-Capable Cellular Networks. In *CCS*, 2005.

[19] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin. A first look at traffic on smartphones. In *IMC*, 2010.

[20] FCC. FCC approves AT&T acquisition of Qualcomm Licenses. 2011.

[21] Y. Go, D. F. Kune, S. Woo, K. Park, and Y. Kim. Towards accurate accounting of cellular data for tcp retransmission. In *HotMobile'13*.

[22] Y. Go, J. Won, D. F. Kune, E. Jeong, Y. Kim, and K. Park. Gaining Control of Cellular Traffic Accounting by Spurious TCP Retransmission. In *NDSS*, 2014.

[23] S. Ha, S. Sen, C. Joe-Wong, Y. Im, and M. Chiang. TUBE: Time-dependent Pricing for Mobile Data. In *SIGCOMM*, 2012.

[24] C. Lever, M. Antonakakis, B. Reaves, P. Traynor, and W. Lee. The Core of the Matter: Analyzing Malicious Traffic in Cellular Carriers. In *NDSS*, 2013.

[25] mobiThinking. Global mobile statistics 2013.

[26] E. R. M. R. Myrto Arapinis, Loretta Ilaria Mancini. Privacy through pseudonymity in mobile telephony systems. In *NDSS*, 2014.

[27] C. Peng, C. Li, G. Tu, S. Lu, and L. Zhang. Mobile Data Charging: New Attacks and Countermeasures. In *ACM CCS*, 2012.

[28] C. Peng, G. Tu, C. Li, and S. Lu. Can We Pay for What We Get in 3G Data Access? In *MobiCom*, 2012.

[29] Z. Qian and Z. Mao. Off-Path TCP Sequence Number Inference Attack-How Firewall Middleboxes Reduce Security. In *S&P*, 2012.

[30] Z. Qian, Z. M. Mao, and Y. Xie. Collaborative TCP Sequence Number Inference Attack: How to Crack Sequence Number under a Second. In *ACM CCS*, 2012.

[31] R. Racic, D. Ma, and H. Chen. Exploiting MMS vulnerabilities to stealthily exhaust mobile phone's battery. In *SecureComm'06*.

[32] S. Sen, C. Joe-Wong, S. Ha, and M. Chiang. Pricing Data: A Look at Past Proposals, Current Plans, and Future Trends. *CoRR*, 2012.

[33] P. Traynor, W. Enck, P. McDaniel, and T. La Porta. Mitigating Attacks on Open Functionality in SMS-capable Cellular Networks. In *ACM MobiCom*, 2006.

[34] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, and T. L. Porta. On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core. In *ACM CCS*, 2009.

[35] P. Traynor, P. McDaniel, and T. La Porta. On Attack Causality in Internet-Connected Cellular Networks. In *USENIX Security*, 2007.

[36] G.-H. Tu, C. Peng, C.-Y. Li, X. Ma, H. Wang, T. Wang, and S. Lu. Accounting for Roaming Users on Mobile Data Access: Issues and Root Causes. In *ACM MobiSys*, 2013.

[37] R. Wang, L. Xing, X. Wang, and S. Chen. Unauthorized origin crossing on mobile platforms: Threats and mitigation. In *CCS*, 2013.

[38] Z. Wang, Z. Qian, Q. Xu, Z. Mao, and M. Zhang. An Untold Story of Middleboxes in Cellular Networks. In *SIGCOMM*, 2011.

[39] Y. Zhou and X. Jiang. Dissecting Android Malware: Characterization and Evolution. In *IEEE S&P*, 2012.