Femtocell Security in Theory and Practice

Fabian van den Broek and Ronny Wichers Schreur

Digital Security, Radboud University Nijmegen
{f.vandenbroek,ronny}@cs.ru.nl

Abstract. Femtocells are low-powered cellular base stations for mobile telephone networks, meant for home use, but still operator managed. They are an increasingly popular solution, with the number of femtocells expected to outnumber the normal cell towers by Q1 of 2013 [1].
However, femtocells also introduce a number of security concerns. Several earlier femtocells have been hacked to varying degree and analyzed. Naturally, the industry has responded and tries to create more secure femtocells.
We provide a first comprehensive analysis of the risks of attacks, given a general femtocell model. This analysis results in two new attacks. We then illustrate some of the dangers by successfully compromising a specific femtocell: the SignaalPlus Plug & Play, sold in the Netherlands by Vodafone.

1 Introduction

In mobile telephony networks such as GSM, UMTS and EV-DO (an American counterpart to UMTS), service is provided through many antennae that each cover a geographic area. These areas are called cells and can range in size based on the transmission power of the signal and the available bandwidth. Within each cell the coverage is influenced differently by local propagation conditions which can result in blind spots where signal reception is so poor that no service is available. To solve this small cells can be created within these blind spots, with a low power antenna that operates on a different frequency from its containing cell.

Small cells can have different sizes, which are usually subdivided into microcell, nanocell and femtocell, from small to smallest. The normal, much larger, cell size is referred to as macrocell. The distinction between the types of small cells is not officially defined, but typically a microcell covers an area the size of a shopping mall or a transportation hub, a nanocell covers a small business or an office floor, and the femtocell a small house or several rooms [1].

Besides the coverage size there is a more important distinction between the femtocell and other small cells. The microcells and nanocells are installed and maintained by the provider and connect directly to the provider's core network, while the femtocell is a consumer-installed (and owned) device and connects to the core network of the provider through the consumer's broadband connection. Naturally this introduces several new security risks for both provider and consumer, since a low-cost device is now placed at the consumer's home, which has the ability to act as an authentic cell tower and connects to the provider's back end over an untrusted channel.

A femtocell device is a small box with a power and Ethernet connector and at least one antenna. Some of the femtocells have GPS onboard, to verify their geographical location. All of them can listen to neighboring cells, in order to run on a non-interfering frequency. Usually femtocells contain a dedicated chip that is specifically made for femtocell devices. These chips consist of a base band processor[1], some cryptographic processor and a general purpose processor. All of the femtocells analyzed so far run some lightweight form of the Linux operating system.

The rest of this paper is structured as follows. Section 2 gives an overview of femtocells within a cellular network. Section 3 gives an overview of the femtocell security model we assume and the most likely attack vectors. In Section 4 we discuss possible attacks offered by a compromised femtocell against the 3GPP security goals for UMTS and LTE. A practical security analysis is presented in Section 5 where we successfully compromise a modern femtocell (the Vodafone SignaalPlus Plug & Play). Finally, we discuss our conclusions and some ideas for future work.

Related work 3GPP, the standardization body for the GSM, UMTS and LTE systems, has specified the use of femtocells within mobile telephony networks. Of these specifications 25-467 [2] and 33-320 [3] are the most interesting, and respectively detail the architecture of and the security architecture of the femtocell (called a Home NodeB or HNB).

Several books have been written on femtocells. "Femtocell Primer" [4] is a very superficial introduction into femtocells, and focuses more on the economic aspects of introducing femtocells. Two other books, "Femtocells: Technologies and Deployment" [5] and "Femtocells: Design and Application" [6] cover femtocells more extensively. They highlight all the technical difficulties in realizing femtocells from an engineering standpoint. Both books contain a small section on security, with only a broad overview of the subject.

Some publications analyze possible security problems that arise when femtocells are introduced in the network [7,8]. Both are theoretical analyses. Tyler et. al [9] show the economic incentives of possible attackers to use a compromised femtocell to DDoS a telecommunications network.

There have also been practical analyses of a physical femtocell device. Indeed several off-the-shelf femtocells have been hacked with varying consequences. In 2010, a research group that calls itself THC (The Hackers Choice) managed to gain root access to the Vodafone Sure Signal femtocell [10]. This proved a very severe security break, based on an easy to guess root password, which allowed interception of phone calls and allowed attackers to request the current session keys form any handset, from the Vodafone back end. A Samsung Femtocell was rooted by a group of researchers from Trustwave's SpiderLabs in 2011 [11,12].

[1] A dedicated processor for signal processing and real-time transmission operations.

We could not find any publications showing the attack capabilities they gained with getting root access to this femtocell.

Researchers from the Technical University of Berlin [13] analyzed the security of a femtocell by Ubiquisys. They manage to break its security and run arbitrary code on the femtocell, which also included the functionality to request session keys for connected phones. They conclude with a rather brief list of possible attacks against the femtocell and the core network with their compromised femtocell. A second publication by the same group [14] presents the method used to break this femtocell and shows that this break compromises all security requirements.

Theoretical and practical research are combined in a publication from researchers in Birmingham together with the group from TU Berlin [15]. In this work they formally verified the authentication in the UMTS and LTE systems using ProVerif, discovered an attack on location privacy, and proved the feasibility of this attack by reprogramming a femtocell.

2 Femtocell overview

The femtocell idea can be applied to many different cellular communication networks, such as UMTS, LTE and EV-DO. Since each of these has its own terminology for network entities, each network also has their own names for the femtocell and the extra network components required for femtocells. For instance, in UMTS the cell towers are called NodeB, so the femtocell is called HNB (Home NodeB). In LTE, on the other hand, femtocells are called HeNB (Home eNodeB). All these different acronyms can make the different specifications difficult to read. Figure 1 shows a femtocell inside a UMTS network. Here a UE (User Equipment, the handset) contains a SIM card and connects to the RAN (Radio Access Network) either via a cell tower, or through a femtocell called a HNB. For a user who connects to the femtocell, the experience should be indiscernible from connecting to regular cell towers. So a running session should be seamlessly handed-over between the HNB and the NodeBs, dependent on signal strength. From the RAN a connection is made to the Core Network of a provider. The SeGW (Security Gateway) is the entity in the provider's core network where the encrypted connection from the HNB over the untrusted Internet connection terminates. The HNB-GW (HNB Gateway) then routes the decrypted traffic inside the provider's core network. The HNB-GW can be combined with the SeGW in a single entity. Communication can be routed to the HSS (Home Subscriber Server), which primarily handles the authentication of SIM[2] cards. There is also a HNB Management Server (HMS), which manages practicalities such as firmware updates and the operational frequencies. The SGSN is also shown as a part of the core network in Figure 1, but this is merely there for completeness

[2] In GSM terminology, SIM card can mean either the physical smart card or the application which runs on it. For UMTS the physical smart card is called a UICC and the application is called USIM. For simplicity we only speak of SIM cards here.

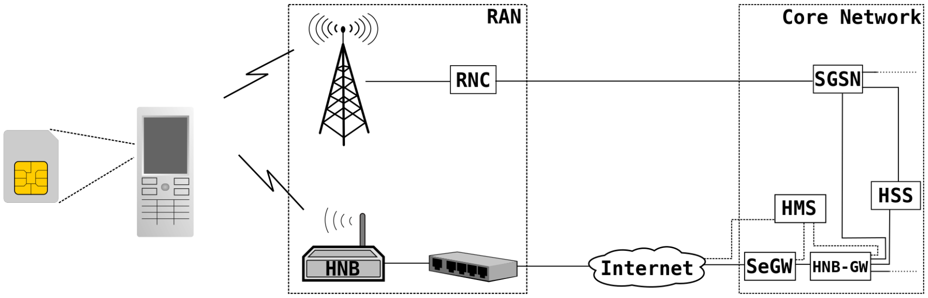

Fig. 1. Diagram of a UMTS network that incorporates a femtocell

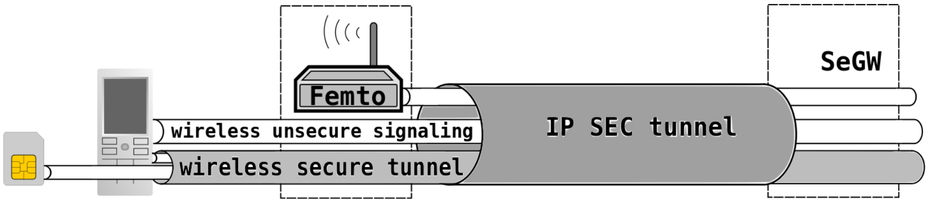

Fig. 2. Tunneled communication between handset and core network over a femtocell

sake, as the cell tower's equivalent of the femtocell's gateway and not important for any of the further discussion.

Again different terminology is used by the Small Cell Forum, a consortium of industry players that advocate the use of femtocells — they actually started out under the name Femtocell Forum. This terminology is also used in most books on femtocells and is meant as a communication network independent generalization of the femtocell idea. They mostly describe the same network entities under a different name. Here a FAP (Femto Access Point) is used to designate the actual femtocell radio unit. The Femtocell gateway is often combined with the security gateway and called the FGW. The HMS is now called a FMS (Femto Management Server) and the HSS is replaced by a more generic AAA (Authentication, Authorization and Accounting) server.

In the rest of this paper we will attempt to avoid any specific terminology, instead we refer simply to femtocell and handset. However in cases where a specific term is needed we will use the 3GPP terminology.

3 The Security model

In earlier cellular systems, such as GSM, the handsets did not authenticate cell towers. This allowed attackers to impersonate these cell towers. Modern cellular networks protect against this attack through mutual authentication between handset and network. The network creates a fresh authentication token based on a sequence number and a shared symmetric key (stored in the provider's core network and inside the subscriber's SIM) for each authentication. Handsets will only connect to cell towers that transmit the correct authentication token. So the provider's core network is authenticated and the handset can assume the cell tower is genuine since it was able to obtain this token. The handset then also responds to a challenge sent by the network, so it to can be authenticated by the network.

A femtocell has a wireless connection to a handset and even before the initial authentication some communication is needed. A femtocell is a device that is supposed to function as a small cell tower and is therefore able to relay the authentication tokens from the core network. Both the connection into the provider's core network and the connection a femtocell makes to handsets can be interesting attack vectors that might threaten the cellular network's security model. Therefore there are several security features advised for femtocells in several specifications [2,3,16].

This section gives an overview on the security model of a femtocell.

3.1 The femtocell security model

Figures 1 and 2 give an overview of the femtocell inside a cellular network. The communication between the femtocell and the core network of the provider needs to be tunneled through an authenticated and encrypted connection. The specifications advice the use of an IPSEC connection between the femtocell and the SeGW, for instance by using IKEv2, which provides authentication based on PKI certificates and integrity and confidentiality on an IP level.

IKEv2 has been formally analyzed and shown to be a secure protocol [17] provided both entities keep their secret key hidden. Naturally the keys inside the femtocell need to be stored securely, for example by placing them on a smart card or inside a TPM (Trusted Platform Module).

Figure 1 shows that the femtocell can contact the management server directly or through the secure tunnel. Both scenarios are presented in the specifications, although the preferred approach is to use the SeGW. For this paper we assume network designs where the management server is placed behind the SeGW and all communication between the management server and the femtocell is thus protected by the IPSEC tunnel. This design seems to make more sense and is also the only behavior we have encountered in the modern femtocells we have investigated.

All the communication between a femtocell and the core network are routed through the IPSEC tunnel. This communication consists of signaling information and user data. Most of these are again inside their own secure tunnel between handset and the core network.

A femtocell has to pass on the authentication messages from both the handset and the core network unaltered for the handset to connect to the femtocell. In this process the session keys between handset and core network are established, which are used to create the secure tunnel between handset and core network (the inner tunnel in Figure 2). These session keys can not be computed or retrieved by the femtocell.

It is possible to design a femtocell in such a way that it also stores the session keys for the secure wireless tunnel. This would seem like a needlessly insecure option, but it does provide some usability benefits, since Internet traffic can then immediately be routed onto the Internet from the user's router, instead of via the provider's back end. This feature is referred to as *local break-out*. We will use the term "local break-out" to refer to a femtocell that was designed with the possibility to store or request the user session keys for the secure wireless tunnel, regardless of the implementation of the local break-out feature. Our practical experiments were on a femtocell model that did not use local break-out, so we assume a model in which femtocells do not support this feature.

Femtocells can be run in two different modes: open and closed. These modes refer to the femtocell's behavior with respect to handsets that do not belong to the consumer. In open mode, the femtocell allows any handset (usually only from subscribers to the same provider) to camp on it. In most cases the subscriber who bought the femtocell can manage the femtocells operating mode and its CSG. The 3GPP specifications allow for two types of femtocells, a CSG femtocell and a non-CSG femtocell [2]. The difference between these femtocells is whether the femtocell or the core network checks if a handset is a member of the CSG. A CSG femtocell maintains the Access Control List of identities (IMSIs) allowed within the CSG, while a non-CSG femtocell is oblivious to the existence of a CSG, all the CSG management is then handled in the core network of the provider.

3.2 Attack vectors

Assuming an attacker has no access to the core network of the provider, the addition of a femtocell into a telco network introduces three new entry points for an attack: the wireless interface, a direct attack on the femtocell device, and an attack on the Internet back haul connection.

The first, the wireless interface, is the same as the standard wireless cellular interface, and so femtocells introduce no new threats compared to the normal wireless interface of the telco network.

The last, the untrusted Internet back haul, delivers a serious threat to the overall security of a telco network. This is mitigated by using a secure IPSEC tunnel, which provides authenticity, integrity and confidentiality.

This makes a direct attack on the femtocell the most viable entry point for an attack, especially since the femtocell also stores the secrets that are needed to set up the IPSEC tunnel.

4 Theoretical security analysis of femtocells without local break-out

This section looks at possible attacks with a compromised femtocell against the security model of UMTS and LTE. So the weaknesses of GSM and fallback attacks to GSM are not considered. With a compromised femtocell we mean a femtocell on which a hacker can execute arbitrary code. This scenario seems likely, since a femtocell is a reasonably low-cost device that is placed in the care of consumers for an extended period of time and which includes a lot of software on a standard execution platform, running Linux. The hacker might not be able to learn the securely stored secrets, e.g. those on a smart card or in a TPM, but he can access the functionality these provide, like signing or encrypting.

We assume a femtocell design that does not receive any user session keys from the provider's core network. So we assume a femtocell without local break-out, which means the attacker is able to listen to, and influence, messages inside the IPSEC tunnel, but not to messages inside the secure wireless tunnel, nor is he able to influence any decisions made in the provider's core network. This is the main difference with most other analyses [7,14,13,10]. We believe it is more realistic to assume a femtocell without local break-out, since the femtocell we investigated does not support it and it seems the most sensible design choice, security wise. Though certainly femtocells with local break-out exist [10,14], which are therefore more interesting targets for attackers, it does not seem unreasonable to assume these devices will be phased out in the future.

The 3GPP standardization organization has specified several security goals for the UMTS and LTE cellular systems [16,18] which expand the security goals that were stated for GSM [19]. We will now see what the impact of a compromised femtocell is on all the goals that could conceivably be influenced by femtocells. In some cases this will add new attacks that were previously impossible. In other cases an already existing attack that is currently hard to perform due to the cost of implementing UMTS/LTE signal processing, could be made easier to implement with a compromised femtocell, because it already handles all the signal processing out of the box. This effectively means the introductions of femtocells can lower the costs of an attack.

User data confidentiality and integrity These two security goals concern the confidentiality and integrity of user data against eavesdroppers and active attackers. Lawful interception is an exception on user data confidentiality.

None of them are weakened by a compromised femtocell, when we assume that no local break-out is implemented in the femtocell, as there is a secure tunnel from the handset to the provider's core network, which is authenticated and provides both confidentiality and integrity. It is infeasible for a compromised femtocell to decrypt or compromise this traffic (assuming strong enough encryption and MACs are used, such as the KASUMI cipher in UMTS). It is also impossible to influence the encryption choice of the network.

Network authentication This security goal was specifically added for UMTS security to mitigate an important weakness of GSM. It aims to protect subscribers from fake cell towers through authentication of the network and is unbroken by a compromised femtocell.

This authentication is done by the so-called UMTS-AKA protocol. The network provides a handset with cryptographic proof of knowledge of a shared secret key and a sequence number to prevent retransmission attacks. The UMTS-AKA protocol was formally analyzed using enhanced BAN logic and shown to provide both authentication and confidentiality [17].

The femtocell never learns the secret key shared between handset (more specifically SIM card) and network and is as such unable to fake a connection to the real network. Retransmission of an authentication token is infeasible because of the sequence number. When a correct UMTS-AKA run finishes, handset and network communicate through a secure tunnel. This makes it impossible for a compromised femtocell to hijack the session without local break-out.

Subscriber identity authentication Subscriber identity authentication is meant to protect the network against unauthorized use by ensuring that the subscriber identity transmitted to the provider is the one claimed.

This security goal is ensured through the mutual authentication of handset and core network. This mutual authentication uses the UMTS-AKA protocol, which was formally analyzed using enhanced BAN logic, and shown to provide both authentication and confidentiality [17]. The authentication itself does not happen on the femtocell, but inside the provider's core network, so insider attacks, such as swapping the authentication tokens inside the network [20], are not feasible from a femtocell without local break-out.

So attacks need to circumvent the UMTS-AKA protocol. A possibility is to place an emergency call at a femtocell and immediately place another call afterwards. An emergency call creates an unauthenticated radiolink, and this link is kept open for the second call. This results in theft of service with a possibly spoofed subscriber identity. However, this threat is detectable by the core network and as such this risk is accepted in the specifications [3]. This attack does not require a compromised femtocell, but could be more easily realized with a compromised femtocell. Due to the detectability the impact is probably small.

Subscriber identity confidentiality Subscriber identity confidentiality comes down to the secrecy of the IMSI number from eavesdroppers and active attackers. This secrecy is already problematic in current networks due to an *identity request* procedure, which causes the handset to respond with its IMSI in plaintext. So, this attack — often referred to as an IMSI catcher attack — is not introduced by a compromised femtocell, though a compromised femtocell does make the execution of an IMSI-catcher attack a lot easier [14].

The specifications also explicitly state that there should not be any relation between the IMSI number and the subscriber's phone number, other than in a database in the provider's back-end. The check whether a handset (or, more accurately, a SIM card) belongs to the CSG — the Closed Subscribers Group, discussed in Section 3.1 — can be made inside the femtocell or within the provider's core network. In the former case a compromised femtocell can uncover the IMSI-phone number relation by adding phone numbers to the CSG, which is standard functionality available to the femtocell owner. The phone numbers need to be translated to IMSIs by the core network and subsequently stored in a CSG femtocell, where they can be uncovered by an attacker. This attack against the subscriber identity confidentiality is more effective than IMSI-catcher attacks, because victims do not need to be connected to, or even near, the attacker. As far as we can tell, this IMSI-*harvest* attack is a new attack, with a higher impact than other attacks against subscriber identity confidentiality.

Signaling confidentiality The signaling messages between handset and network should remain confidential. Since we assume a femtocell without local break-out, only the signaling that is transmitted outside of the user session tunnel is subject to confidentiality breaches. Since this concerns all unencrypted messages on the wireless link, no new weaknesses are introduced by a compromised femtocell, although some attacks are easier to implement with one.

The most prominent attack here is IMSI catching, which is detailed under the security goal *Subscriber identity confidentiality.* Another attack vector lies in the paging channel. Handsets listen to this channel to see if they have incoming transmissions, by looking for occurrences of their IMSI or, more frequently, their TMSI (a temporary pseudonym for their IMSI). This could lead to a traffic analysis where all incoming transmissions for subscribers connected to a compromised femtocell can be revealed.

Signaling integrity An attacker should not be able to alter signalling messages between handset and network. A compromised femtocell introduces the attack that makes it possible to alter unencrypted signallng messages.

The user session tunnel guarantees integrity of messages, so any attacks against signaling integrity, have to be made on untunneled signaling. Attacks against signaling integrity can lead to DoS attacks, which are discussed in the section on Availability shown below. Other possible attacks are to fake paging messages to handsets — which cause a handset to indicate incoming transmissions, when there are none — or abuse of the broadcast messages — such as fake alert messages of the Public Warning System (PWS) [21].

Subscriber location privacy and untraceability The current or earlier location of a subscriber should not be derivable from transmissions on the air interface.

The recent location privacy attack from Birmingham [15] unveiled a weakness in the UMTS protocol that can be used to break subscriber location privacy. In short, cell towers send an *authentication request* message to a mobile phone. This message contains both a proof that the network knows the SIM card's secret key and a sequence number needed for freshness. The mobile phone responds with an error message if the proof of knowledge of the secret key is incorrect, or with a different error message if the sequence number is incorrect. So by replaying any, earlier recorded, correct *authentication request* message for a specific phone, an attacker can see if the target phone is in his current cell.

Their attack was implemented on a femtocell as a proof-of-concept. The implementation showed this attack is indeed viable from any 3G cell tower ranging from femtocell to macrocell.

This attack is also viable from a compromised femtocell without local break-out, since the correct *authentication request* messages are sent plain text from the provider's core network, as no session key is yet established, and can always be replayed without access to the session secrets.

Availability Attacks against availability, commonly known as DoS, are considered in the UMTS/LTE specifications [22], but availability is not officially stated as security goal [16,18]. DoS attacks performed from a femtocell can be subdivided into two categories: DoS attacks towards the subscriber and DoS attacks towards the provider.

DoS attacks towards the subscriber are trivially possible by blocking any incoming or outgoing data transmissions on a subscriber camping on a compromised femtocell. Another method to perform a DoS attack is to send malformed packages to the handsets, which attempt to compromise their base-band stack. This could be done at very low layers of the protocol (for example by changing something in the waveforms), below the layer with the integrity checks of the secure tunnel, or by attacking all the layers in the untunneled signaling messages, such as the broadcast and paging messages. This process of creating malformed packets is called *fuzzing*. To our knowledge this attack has not been attempted on UMTS base band stacks. However, several fuzzing attacks against GSM have shown that older (GSM) base band stacks are vulnerable to this [23].

Another DoS attack that was possible with earlier femtocells [24,14], is no longer possible on a femtocell without local break-out. In this attack the IMSI detach message of a camped handset is faked. This will cause the network to assume a phone has been switched off and therefore hold all inbound transmissions to this handset. However, this attack only works as long as a handset is connected to the compromised femtocell, and the femtocell needs local break-out to perform it.

DoS attacks towards the provider's core network also seem possible. The most obvious point would be to attack the SeGW, since this entity sets up the IPSEC connections, and therefore needs to do many calculations in order to send and verify received cryptographic messages. If many attacking machines attempt to set up an IPSEC connection with the SeGW it will get overloaded and the connections between the SeGW and genuine femtocells will suffer. An attacker would not need to compromise a femtocell for this, though access to the secrets needed to set up the IPSEC tunnel can make this attack more effective, by causing more computations in the SeGW. Whether it is possible to DoS other entities in the provider's core network, such as the *AAA/HSS*, is hard to predict. As we discussed, there have been several successful fuzzing attacks against handsets. So it would seem logical to assume that the base band stack on network equipment is also vulnerable when handling packets just outside of the specifications. Some attacks against the core network were found by a private security company [25], which seems to support this assumption, but it remains impossible to test without access to a test network or by possibly harming the real network.

Table 1. Overview of successful attacks on femtocells; the last entry is the femtocell we attack in this paper

	Vendor	Type	Weakness	reference
1	Sagemcom	Vodafone SureSignal	Guessable rootpw	[10]
2	Samsung	Verizon SCS-24UC4 & SCS-2U01 & Sprint Airave	Adjustable boot loader	[12,11]
3	Ubiquisys	SFR Home 3G	Insecure update procedure	[13,14,15]
4	Sagemcom	Vodafone SignaalPlus Plug&Play	Insecure recovery mode	

5 Practical security analysis of the Vodafone Plug&Play femtocell

The femtocell itself needs to be a hardened device, since it contains credentials to authenticate to the provider's core network and is placed at the subscriber's home, but it should still be under the management and control of the provider. For a practical security analysis on a modern femtocell we looked at the Vodafone SignaalPlus Plug&Play, the first commercially available femtocell in The Netherlands, available for 80 euros. We first give a high level overview of our attack and discuss its nett effect. We then provide the details for the interested reader.

Overview of the attack We were able to read out the unencrypted memory of the femtocell, which provided all the secrets needed for our attack. It proved possible to reboot the femtocell in an insecure recovery state, by sending a command over the ethernet connection on a TCP port. The firewall that runs on the femtocell only opens up this port after a secret port-knocking sequence is completed. Once in recovery mode the femtocell has SSH enabled and attempts to retrieve a file via a tftp session to a local network address, which it then executes. We provided the femtocell with a file that gave us a root login on its SSH prompt.

The recovery mode of the femtocell runs a different Linux version than the normal mode. From the recovery kernel we can mount all the other partitions, but we cannot get the femtocell into operation, since most program won't run form the recovery kernel version. So getting the femtocell in operation would require rewriting most of the binaries for this specific kernel version. Also, this implementation would invariably need to be tested, which could result in some non standard behavior that might be observable inside the Vodafone back end.

However, we were able to compile programs that run on the femtocell in recovery mode. This means we can run arbitrary code on this femtocell, in essence this breaks the security model as detailed in Section 4.

The details We first attempted attacks that were successful on older femtocell models (summarized in Table 1):

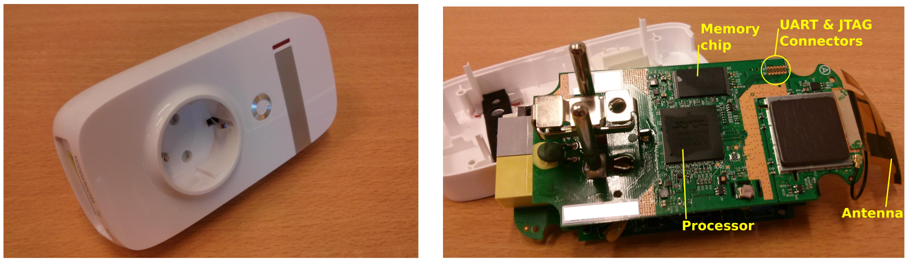

Fig. 3. The Vodafone SignaalPlus Plug&Play femtocell.

1. There was no SSH running on our femtocell, so easy guessable root passwords where out.
2. Researchers looked for the differences between the source code made available by Samsung[3], to which they were obliged under GPL, and the stock open source versions. They found a key that allowed them to enter the boot loader's menu via the serial port. In our case the femtocell also uses GPL code, but the code placed online by Vodafone was not the same as the code that runs on the device (which we could uncover by dumping the memory chip). Although this is a clear violation of the GPL license, it did hinder our efforts.
3. Holding the reset button does not prompt our femtocell to connect to an insecure update server, like the Ubiquisys did.

So, all previous attacks failed. The Vodafone SignaalPlus Plug&Play, shown in Figure 3, is manufactured by Sagemcom. Inside the casing (which can be removed without any physical counter measures) there are two connected PCBs, with a Percello 6000 chip and a flash memory chip. The JTAG connectors are logically disabled[4], but there are active UART connectors that show a console log during the boot sequence.

The Percello 6000 chip has a built-in TPM that presumably contains the secrets needed to set up the IPSEC tunnel. The boot logs also show that the integrity of the code which runs on the femtocell is verified: the hashes of some files are compared with signed hashes from the TPM.

The data on the flash memory chip is unencrypted and stored in an UBI format and is divided in several volumes that we were able to dump through direct access to the memory chip. Nearly every volume needed for normal operation has an exact duplicate labeled either with an *A* or *B* post-fix, which seems built-in redundancy (for instance in case an update fails and corrupts the file system).

[3] The source code was made available via the webpage `http://www.samsung.com/global/business/telecommunication-systems/resource/opensource/femtocell.html`

[4] JTAG (Joint Test Action Group) is an industry standard for debugging access to embedded processors and printed circuit boards.

The other partitions also have an exact duplicate, but now with a *_BKP* postfix. Only the recovery partition has no duplicate.

The femtocell runs a firewall that blocks most ports. However, a port-knocking daemon also runs from the *RFS_A* partition. We set up the femtocell on an internal network, without outgoing Internet connection. On this internal network we ran our own DHCP server and DNS gateway. After sending packages to the femtocell in the order of the porting-knocking sequence, the final port opened in the firewall for a TCP connection. Reverse engineering the binary that listens to this newly opened port revealed two commands: "reboot recovery" and "switch bank". After the command "switch bank" was sent to the newly opened port, the femtocell boots from the B partitions (or back from the A partitions). More interestingly the "reboot recovery" command causes the femtocell to boot into a recovery mode.

A trace from the WireShark network protocol analyzer showed that in recovery mode the femtocell attempts establish a tftp session to the fixed IP address 192.168.1.1 and requests a file called `femto3xx/originalsin`. The *LINUX_R* volume contains the recovery kernel, which is booted in recovery mode, and a compressed recovery file system. This filesystem contains a file `adam` that is called immediately after the boot procedure. This file proved to be a simple script in the execline syntax that tries to tftp the file `femto3xx/originalsin` into a temporary file `eve`. This `eve` file then has the executable bit set and is executed. Since `adam` runs with root privileges, the attack file that we offer for the tftp session can put our public key in the SSH `authorized_keys` file of the root account. This gives us root access through SSH on the recovery mode of the femtocell.

We were able to replicate the attack on multiple femtocell devices of the same version. Through the shadow files we found that the root password is the same for every device we gained access to. Running John the Ripper on the root password hash yielded no results.

Using a MIPS compiler we can compile programs that run onto the femtocell, and this gives us arbitrary code execution on the femtocell.

6 Future Work

A compromised femtocell without local break-out offers some attack possibilities discussed in Section 4, which should be examined further. Most prominently these are the integrity attacks against the untunneled signaling messages that could offer up new attacks. Also, a compromised femtocell can make fuzzing attacks over UMTS protocols against handsets possible, which to our knowledge have not been attempted before.

We also see some ways to improve our attack against the Vodafone SignaalPlus Plug&Play femtocell. It might be possible to reactivate the JTAG connectors. This would allow a degree of control on the processor that our current attack does not provide.

Our attack could also be extended in using the TPM as an oracle, in order to analyze the data sent through the IPSEC tunnel which are not part of the 3G traffic, so all the management data. We are able to execute arbitrary code on the femtocell, which makes this approach possible, but in the interest of time we were unable to perform this attack.

7 Conclusion

We have provided the first comprehensive security analysis of a femtocell without local break-out in Section 4. We have shown that a compromised femtocell enables attacks that directly impact several security goals:

- Subscriber identity confidentiality
- Signaling integrity
- Availability

Several attacks already exist without a compromised femtocell, but we argue that some of these are much easier to exploit with the use of a compromised femtocell. This resulted in easier implementation of attacks against several security goals:

- Subscriber identity authentication
- Subscriber identity confidentiality
- Signaling confidentiality
- Availability

Several attacks using older model femtocells with local break-out, are not possible in our model of a femtocell without local break-out. Of these, the eavesdrop attack on subscriber data probably has the most impact. So, the security of a cellular network with femtocells is improved when the femtocells do not support local break-out; in essence the provider places less trust in a femtocell.

Our analysis resulted in two new attacks, which to our knowledge were not published earlier: (i) the IMSI-harvest attack discussed in the section on Subscriber identity confidentiality (Page 8) and (ii) fake Public Warning System messages, discussed in the section on Signaling integrity (Page 9).

We also show a practical attack on a modern femtocell without local break-out. A dump of the code of the femtocell enabled us to learn the port-knocking sequence that allows the femtocell to go into an insecure recovery mode, which retrieves a file and executes it. With a couple of days of effort, we were able to gain root access to this device and able to execute arbitrary code on it. We gain fewer capabilities than previous hacks of older femtocells (which did implement local break-out). Our femtocell was also secured against earlier known attacks.

We made some interesting observations while examining the femtocell. First of all, in accordance with the GPL, Vodafone provides a link to source code. However, the provided source code is not the code that actually runs on the femtocell. It appears to be code meant for an older version of different hardware by Alcatel-Lucent, instead of the current version by Sagemcom. This is clearly a violation of the GPL and it forced us to dump the contents of the memory chip for analysis.

Secondly, it seems strange to disable SSH access, but to allow access to the femtocell through the secrecy of a port-knocking sequence, which is poor security, since the secret sequence cannot be stored securely. However, the benefit of the port-knocking defense is that this will only work locally, since most devices will be placed in a NAT environment in a subscriber's home, so the router would already block most ports. SSH on the other hand might be accessible over the Internet. This would have been especially worrying, since we found that all devices of this type we bought had the same root password.

Both our theoretical and practical analysis suggest the security of femtocells is improving. None of the weaknesses from earlier models were present in the new femtocell. Though the main improvement is that the providers place less trust in the femtocell devices, because the femtocells do not provide local-breakout. One should always assume that a femtocell will eventually fall under control of an attacker, so the less trust that is placed in the femtocell, the better. Femtocells without local break-out are a definite improvement, as are femtocells that do not check the membership of the closed subscribers group themselves.

However, femtocells with local break-out are still available on the market and as long as these can connect to the core network, femtocells without local break-out add little security. Even with these femtocells without local break-out some attacks remain possible when a femtocell gets compromised, though these attacks typically have a lower impact.

Responsible Disclosure and Acknowledgements

We informed Vodafone Netherlands of our findings. They informed us that recent models of their femto cell do not expose the recovery mode. We could confirm that our attack indeed no longer works on these models.

Thanks to Joeri de Ruiter and Roel Verdult for their assistance in the practical security analysis.

References

1. Small Cell Forum. Homepage of the Small Cell Forum. `http://www.smallcellforum.org/`, visited in February 2013.
2. European Telecommunications Standards Institute, France. *Universal Mobile Telecommunications System (UMTS); UTRAN architecture for 3G Home Node B (HNB); Stage 2*, 2012. 3GPP TS 25.467 version 11.0.0 Release 11.
3. European Telecommunications Standards Institute, France. *Universal Mobile Telecommunications System (UMTS); LTE; Security of Home Node B (HNB) / Home evolved Node B (HeNB)*, 2012. 3GPP TS 33.320 version 10.5.0 Release 10.
4. David Chambers. *Femtocell Primer (2nd Edition).* Lulu Enterprises Inc., 2010.
5. Jie Zhang and Guillaume de la Roche, editors. *Femtocells: Technologies and Deployment.* John Wiley & Sons, Ltd, 2009.

6. Michael Ruggiero Joseph Boccuzzi. *Femtocells: Design & Application*. McGraw Hill Professional, 2010.
7. R. Rajavelsamy, Jicheol Lee, and Sungho Choi. Towards security architecture for home (evolved) nodeb: challenges, requirements and solutions. *Security and Communication Networks*, 4(4):471–481, 2011.
8. Chan-Kyu Han, Hyoung-Kee Choi, and In-Hwan Kim. Building femtocell more secure with improved proxy signature. In *GLOBECOM IEEE*, December 2009.
9. Vicente Segura and Javier Lahuerta. Modeling the economic incentives of ddos attacks: Femtocell case study. In *EISP 2010*. Springer US, 2010.
10. THC. THC website detailing an attack against a Vodafone SureSignal femtocell. `http://wiki.thc.org/vodafone`, visited in February 2013.
11. Trustwave. Announcement of the samsung femtocell. `https://www.trustwave.com/pressReleases.php?n=012810`, visited in March 2013.
12. Zack Fasel and Matthew Jakubowski. Website detailing how to root the samsung femtocell. `http://rsaxvc.net/blog/2011/7/17/Gaining%20root%20on%20Samsung%20FemtoCells.html`, visited in March 2013.
13. Ravishankar Borgaonkar, Kevin Redon, and Jean-Pierre Seifert. Security analysis of a femtocell device. In *SIN '11*, New York, NY, USA, 2011. ACM.
14. Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. Weaponizing femtocells: the effect of roque devices on mobile telecommunication. In *NDSS '12*. The Internet Society, 2012.
15. Myrto Arapinis, Loretta Mancini, Eike Ritter, Mark Ryan, Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. New privacy issues in mobile telephony: fix and verification. In *CCS '12*, New York, NY, USA, 2012. ACM.
16. European Telecommunications Standards Institute, France. *Universal Mobile Telecommunications System (UMTS); 3G Security; Security Principles and Objectives*, 2001. 3GPP TS 33.120 version 4.0.0 Release 4.
17. European Telecommunications Standards Institute, France. *Universal Mobile Telecommunications System (UMTS); Formal Analysis of the 3G Authentication Protocol*, 2001. 3GPP TR 33.902 version 4.0.0, Release 4.
18. European Telecommunications Standards Institute, France. *Digital cellular telecommunications system (Phase 2+);UMTS;LTE;3G security;Security architecture*, 2013. 3GPP TS 33.102 version 11.5.0 Release 11.
19. European Telecommunications Standards Institute, France. *Digital cellular telecommunications system (Phase 2+); Security aspects*, 1998. EN 300 920 / GSM 02.09.
20. Joe-Kai Tsay and Stig F. Mjølsnes. A vulnerability in the UMTS and LTE authentication and key agreement protocols. In *MMM-ACNS'12*. Springer-Verlag, 2012.
21. GSMA. Mobile network pws and the rise of cell-broadcast. `www.gsma.com/mobilefordevelopment/wp-content/uploads/2013/01/Mobile-Network-Public-Warning-Systems-and-the-Rise-of-Cell-Broadcast.pdf`.
22. European Telecommunications Standards Institute, France. *Universal Mobile Telecommunications System (UMTS); 3G Security; Security Threats and Requirements.*, 2001. 3GPP TS 21.133 version 4.1.0 Release 4.
23. Collin Mulliner, Nico Golde, and Jean-Pierre Seifert. Sms of death: From analyzing to attacking mobile phones on a large scale. In *USENIX Security Symposium*, 2011.
24. Sylvain Munaut. IMSI detach DoS, April 2001. `http://www.blackhat.com/presentations/bh-asia-01/gadiax.ppt`.
25. P1Security. website detailing a fuzzing product for telco core-networks. `http://www.p1sec.com/corp/products/p1-telecom-fuzzer-ptf/`, visited in March 2013.