



Embedded security you can trust

Standards Watch

Elliptic on security standards activity

Issue 7 – Winter 2010

New Attack on GSM Legacy Encryption Makes GSM Networks Vulnerable, Points the Way to New Attacks

A presentation by Karsten Nohl and Chris Paget at the 26th Chaos Computer Congress in December was picked up and widely reported by the popular press. In their presentation, Nohl and Paget make a plea to GSM operators worldwide to enhance the security of confidentiality of voice communications on the air interface. The attack itself is not terribly novel, building as it does on well known cryptanalytic techniques and results for the A5/1 cipher used in GSM networks. However, it is notable for a few reasons: first, the basic approach makes clear that 64 bits is insufficient for encryption key sizes in modern systems; second, a mistake in the protocol design used in GSM makes the keys used for A5/3 encryption (Kasumi based algorithms) vulnerable in some scenarios; and finally it demonstrates just how accessible high powered cryptanalytic tools are today (and how much more so they will become).

At the heart of the attack described by Nohl and Paget is a modification of a technique called “rainbow tables” for precomputing key data for the A5/1 cipher. A5/1 was designed as a very compact, high performance algorithm in the late 1980s. The rainbow tables are very large at around 2 TB (terabytes), but that amount of storage is easily within reach of anybody with \$200 for inexpensive SATA disk drives, at this writing. The data in the tables was computed exploiting the vast computing power available on current generation consumer video adapter graphics processing units (GPUs) and the video game processor (“Cell” computer) found in the Sony Playstation. The tables were computed in about three months using a cooperative distributed computing effort with donated processor cycles. (The design of A5/1 lends itself to very efficient low gate count implementations which was important at the time it was designed, but this same fact would also allow a cryptanalyst - or attacker - to devise highly optimized custom hardware implementations in FPGA or a low-cost older generation IC technology that could be many times faster than the software-based implementation used by Nohl and Paget.) The tables themselves are now being distributed on the peer-to-peer (P2P) BitTorrent network. The tables are then used with a fast A5/1 implementation to do a rapid search of the keyspace for the encryption keys for a particular phone call. The system does not operate in realtime, but a recorded call can be decrypted in several minutes. All-in-all, this is a very modern attack that combines cloud computing with software running

In this Issue

Welcome to the latest edition of Standards Watch. This issue discusses vulnerabilities. The end of 2009 saw reports of new attacks on a legacy cipher in the GSM family of protocols, but might best be described as a sensationalized rallying cry to move quickly to better security for wireless networks. We discuss the implications to 3GPP implementors. Second, a recently discovered practical vulnerability was found in the SSL and TLS families of secure communications protocols. While not unheard of, it is always surprising to find an issue like this in a mature and widely deployed standard. We provide our recommendations for what immediate response is called for, especially in the context of embedded implementations.

Finally we wrap up the issue with a quick look at what is new and upcoming in security in the 3GPP wireless standards.

Quick Note: IEEE has ratified the 802.1AR Secure Device Identity standard. An upcoming issue will examine 802.1AR in detail.

on video processors to create an inexpensively realizable exploit. While weaknesses in the design of A5/1 are exploited to reduce the computing effort required, it is also clear that small key encryption algorithms (those that use 64 bits keys, or 56 bits in the case of DES) are within range of practical short-term attacks using current generation technologies.

The attack also takes advantage to several features of the system to make it faster to implement, and also to go after other parts of the GSM system. While the attack needs some “known plaintext” to implement, this is readily available in network protocols. Packets that occur in regular and predictable points in the protocol sequence are a common feature of many protocols that create known plaintexts. A feature of GSM networks that is still common today is that decryption will fall back to legacy modes if the basestation requests the mobile station (handset) to do so. This fact, together with what can be viewed as a mistake in the system design, allows the A5/1 exploit to reveal the key for the A5/3 (Kasumi) cipher. The same key is used for both A5/1 and A5/3 ciphers, a poor design in any security system. A rogue basestation can associate with a handset (the handset chooses the strongest signal), and require use of A5/1. If it supplies the same nonce supplied by another basestation to initiate a call protected using A5/3 (it can learn this by monitoring the real call setup), the derived key used for A5/3 will be the same as the key used by A5/1. Breaking the A5/1 key also reveals the A5/3 key.

How realistic is it to set up a rogue basestation? Today, it can be done using open source code (there are entire packaged software distributions for doing this) on low cost hardware. Software defined radio chips are increasingly available, and there are companies that distribute the required RF hardware that connect to it over USB ports for about \$1500. Someone who wants to build their own hardware could do so using commercial chips for less than \$100 a copy. Rogue WiFi access points were common in the early days of 802.11 networks. It is not unrealistic to believe that the same will be true with GSM basestations, particularly where there are opportunities to exploit the information gleaned for financial gain. The system design exemplified by the handset's willingness to associate with any basestation that will talk to it also highlights one of the common weaknesses in carrier network design. Carriers greatly care that they control whose data they're willing to carry (they bill customers for that, after all) but customers are not protected by the assurance that they are in fact talking to the network they think they are. This is just plain shoddy system design, but it is repeated over and over again. Thankfully this practice is starting to disappear (GSM's 3GPP system replacement uses mutual cryptographic authentication, for example), but it is still not uncommon to see similar asymmetric protection assurance levels in many new domains.

While it is unfortunately true that these legacy GSM ciphers and network protocols will be with us for some years to come, it is also true that there is reason to believe that the situation will eventually improve. 3GPP's latest data confidentiality, integrity and authentication schemes have benefited from lessons learned from their own mistakes, as well as 20 years advancement in the theory and practice of data security generally. See our last article in this issue for a look ahead.

New Vulnerability Discovered in the SSL/TLS Protocols

In November of last year, Marsh Ray and Steve Dispensa of Phonefactor announced the discovery in August of a new vulnerability in the SSL protocol and its successor TLS. The disclosure of the vulnerability and workarounds for it was prompted by independent discovery of the same issue by Martin Rex of SAP. The latest and most common version of this family of protocols is SSL version 3.0, a widely implemented and publicly released standard developed by Netscape. TLS is an open standard defined by IETF in RFCs 2246 (version 1.0), 4346 (1.1) and 5246 (1.2).

The new vulnerability is a serious issue for two reasons. Firstly, SSL and TLS are arguably the most popular secure communications protocols in existence, used to secure payment transactions on

the Internet, for VPNs and secure access to email over the Internet, and a range of other applications. Secondly, because the flaw is in the protocol itself, every full implementation is vulnerable. So, how much cause for concern is there in this?

To answer this question, the nature of the problem needs to be understood. The simplest description is that the protocol is susceptible to a man-in-the-middle (MITM) attack during renegotiation of the security state of the session. Renegotiation is allowed in order to update or age-out session keys, to upgrade the security parameters of the session, or frequently to convert an anonymous session into one that requires clients to authenticate using cryptographic certificates.

It is this last use that creates the problem. Many web servers that require SSL for some pages start with an anonymous session. Later requests for different pages may now require the client to authenticate using certificate. This triggers the server to prompt the client to renegotiate the session using its certificate. One way for an interloper to accomplish a successful attack is to intercept the initiation of the anonymous session from the client. It then initiates its own anonymous SSL session with the intended server, requesting a page for which client certificate is required and for which it has no certificate. The server initiates renegotiation. The MITM now replays the original client hello and the remainder of the session negotiation messages inside the anonymous session it previously created. The client certificate is one of the forwarded messages: once it has been sent, the request the interloper made is considered authenticated by the server, using the client's certificate. Mission accomplished. Neither client nor server can detect that the interloper has even been involved. The client session will continue through successful negotiation, the interloper simply steps out of the way and the server is never any the wiser.

The underlying cause of the problem is that the new session is not cryptographically bound to the old session using a session identifier that assures that the same endpoints are involved in both sessions. IP addresses or any other network parameter do not fill this role since they are not part of the authenticated data for the protocol, and in any event can change through the life of a session (the so-called superproxy problem). Ultimately, the cure is to use an authenticated binding between the original session and the renegotiated session. This requires a new message in the protocol, and because of the wide deployment of SSL, the change must be backwards compatible and optional for a very long time. (SSL version 2 has been obsolete due to its own vulnerability for more than ten years, but support for it is only now being dropped from common implementations.) So for the next several years, implementations will need to be forgiving of peers they interact with that do not implement the augmented protocol.

How large a problem will this be in practice? Implementation of a successful attack requires the attacker to be able to arrange for traffic to be forwarded through it on the way to the server. This is often difficult to accomplish. The fact that all clients and servers are affected regardless of what technology is implementing the protocol means that vendors of common clients and servers will have solutions ready to deploy quickly - many have already started distributing solutions. For embedded server devices, renegotiation is seldom required. Where practical it should be simply disabled. When client certificate authentication is required, the server should request it during the initial session negotiation, which assures that it is protected by the authenticator provided in the finish message at the end of session negotiation.

So to sum up, the problem is serious and system manufacturers need to start planning how they'll address it. But there are immediate workarounds that can avoid the vulnerability using careful configuration in many cases. In those cases where it is not an option to eliminate session renegotiation, protocol updates necessitating software or firmware updates will be the only option. Customers of Elliptic hardware or software products that have implemented their own versions of SSL and/or TLS protocols should examine their implementations carefully to assess their vulnerability to the issue (nothing in Elliptic products will fundamentally impair the ability of

customers to respond to the issue in their implementations). Elliptic customers that need assistance with vulnerability assessment or implementation updates should feel free to contact us.

New Security Algorithms in Upcoming 3GPP Release 9 and LTE-Advanced

Following hot on the heels of Release 8's new security SNOW 3G algorithms for LTE wireless networks, 3GPP has published its definitions for the 128-EEA2 and 128-EIA2 AES-based algorithms as part of the 3GPP Release 9 specifications. These new algorithms are based around 128 bits key-size ciphers, so will be invulnerable to the kinds of attacks our first article discussed for the foreseeable future. Elliptic has responded with a range of solutions suitable for basestations, femtocell and picocell, and mobile terminal environments. The new algorithms use AES in counter mode to implement payload privacy in 128-EEA2, virtually mirroring the use of SNOW 3G in 128-EEA1. Integrity protection is implemented using AES in CMAC mode, essentially an application of CBC-MAC mode for data integrity. The 128-EIA2 algorithm derives two sub-keys from a single base key to protect the stream.

Basestation and picocell implementors and manufacturers will want to pay particular attention to their system designs to assure that their bandwidth targets can be met with minimal overhead and complexity in their systems. Both 128-EEA2 and 128-EIA2 are used for both control plane messages, and these payloads may be expected to be small. Thus system designs need to account for large numbers of packets with accompanying security data comparable in size to that of the payloads moving across their bus. User plane data is ciphered independently using 128-EEA2. World-market designs need to provide both AES and SNOW 3G algorithms sets to remain relevant. To address the significant bandwidth demands of these new algorithms in basestation applications, Elliptic has announced the second generation of its sCOPE product line including flow-through support for both sets of algorithms. The PLP-101f streaming engine provides a cost effective solution for customers integrating full LTE support in their basestation product lines now, with support for 128-EEA1, -EIA1, -EEA2 and -EIA2 LTE algorithms as well as legacy support or UMTS UEA2 and UIA2 algorithms. Similar support has been announced across Elliptic's entire product line for wireless handset and terminal devices, picocell and femtocell products and new integrated basestation solutions. These are complemented by a full range of wireless networking solutions for WiFi, WiMAX and wired backhaul applications over IPsec.

Looking forward, it is now agreed that LTE-Advanced will be based on 3GPP Release 10. New algorithms to support key Asian markets will be defined in Release 10. As always, Elliptic customers can count on us to be present with solutions addressing full spec compliance by the time Release 10 is finalized.

Contact Elliptic

To discuss anything you've read here, or your particular security needs, contact Elliptic Technologies at:

62 Steacie Drive, Suite 201
Ottawa, ON K2K 2A9, Canada
Phone: +1 613 254-5456
Fax: +1 613 254-7260

www.elliptictech.com
Email: info@elliptictech.com