Secure Location Verification

A Security Analysis of GPS Signal Authentication

Georg T. Becker[1,2], Sherman C. Lo[3], David S. De Lorenzo[3], Per K. Enge[3], and Christof Paar[1]

[1] Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany
[2] University of Massachusetts Amherst, USA
[3] Stanford University, USA

Abstract. The use of location based services has increased significantly over the last few years. However, location information is only sparsely used as a security mechanism. One of the reasons for this is the lack of location verification techniques with global coverage. Recently, a new method for authenticating signals from Global Navigation Satellite Systems(GNSS) such as GPS or Galileo has been proposed. In this paper, we analyze the security of this signal authentication mechanism and show how it can be used to establish a secure location verification service with global coverage. This new security service can be used to increase the security of various different applications, even if they are not directly connected to navigation or positioning.

1 Introduction

The use of location based services has grown tremendously in recent years. One of the reasons is that the GNSS receivers have become very small and cheap. Many mobile phones, for example, already have GPS capability. There has also been research in the area of location based access control, mainly for wireless networks. However, there has been a lack of technologies that can provide location verification services with global coverage. But location has great potential to be used as a cryptographic primitive. In many cases, a communication partner can be identified primarily by its location. In other cases the location can be used as an additional authentication parameter to increase the strength of an authentication protocol. In [1] the idea of using GPS to establish a location verification service was first introduced. However, this approach relied on the fact that the verifier as well as the user both have a trusted GPS-authentication device. Furthermore, this approach is vulnerable to spoofing, especially since the selective availability was turned off. Recently, Lo et al. introduced a new way to authenticate GPS signals in [2] called SAGA. In this paper we analyze the security of this new signal authentication mechanism and evaluate its usability. The advantage of this mechanism is that it does not only increase the security of location *self-verification*, but it can also be used for location *verification*. Previous approaches of location verification either relied on trusted devices or on bidirectional systems with only local coverage such as distance bounding [3]. In SAGA

on the other hand, the location and time of the reception of any GPS signal can be securely determined. This enables a location verification service that does not need any trusted devices and that does not need to have a bidirectional communication with any location service provider. Therefore, SAGA can be used as a building block to set up different secure location based services, ranging from traditional applications such as secure positioning and secure tracking to new applications such as location based access control. The main difference between this paper and [2] is that in [2] SAGA is described from a technical and navigational point of view. It includes a proof of concept implementation and complexity estimations but does not explain any possible attacks on the system nor states any security assumptions. In this paper we will look at SAGA from the computer security point of view. The second chapter is aimed to provide enough information so that non-navigation experts can understand the functionality of SAGA and our security analysis. In the third chapter we then come to the security analysis of SAGA by explaining the possible attacks on the system. From this threat analysis we then derive the security assumptions under which the system is secure. The conclusions from this security analysis are drawn in the last section. As location verification is not commonly defined in the literature yet, we start this paper with a definition of location verification and location self-verification: In secure *location verification from A to B*, B can be sure that entity A was at location L_A at time t. In secure *location self-verification for A*, A can be sure that A was at location L_A at time t.

2 Secure Authentication for GNSS Applications (SAGA)

In this section, we give an overview how Secure Authentication for GNSS Applications (SAGA) works. A more technical description of SAGA with test results can be found in [2]. In this paper we explain SAGA using GPS, but as other GNSS systems such as GALILEO or COMPASS work similarly, they can be used with SAGA as well.

GPS background

In GPS, the position is determine by measuring the arrival times of signals from different GPS satellites. With these arrival times, pseudo ranges between the receiver and the satellites are determined. Trilateration is used to calculate the position of the receiver out of these pseudo-ranges. Signals from four different satellites are needed to solve the trilateration equations, three to determine the 3-dimensional position and one to determine the accurate time. In the current GPS constellation (typically 24-32 satellites), all satellites transmit signals on at least two frequencies: L1 and L2 (at 1575.42 MHz and 1227.60 MHz, respectively). The GPS satellites transmit a civilian signal on the L1 frequency. This C/A-code sequence (for Coarse Acquisition) is publicly available free-of-charge to any user worldwide. GPS satellites also transmit a secret military signal on both L1 and L2. This P-code (for Precision) sequence is encrypted to deny access to

unauthorized users, becoming the P(Y)-code. As all satellites transmit on the same frequency, code-division multiple access is used to ensure that the satellites do not interfere with each other. Figure 1 illustrates the GPS signal structure. The 50 Hz data signal gets added (XOR) with the 1.023 MHz C/A code sequence.

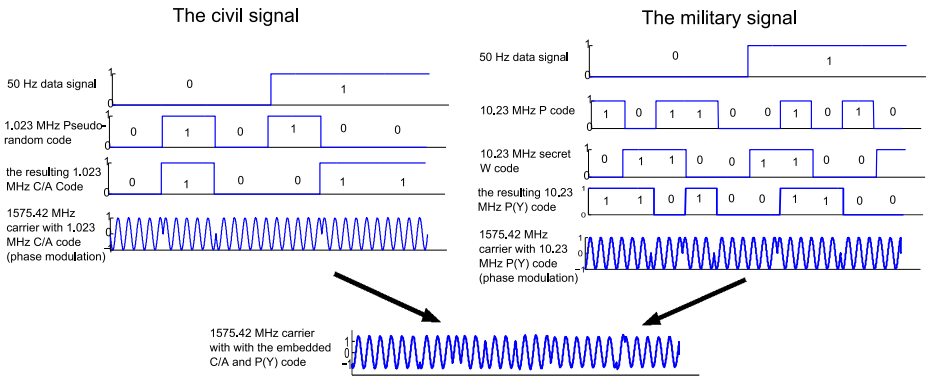

Fig. 1. The GPS signal structure on the L1 frequency.

This C/A code is a pseudo-random code sequence with a length of 1023 bits (also called chips). Each satellite has its own unique pseudo-random code sequence. The resulting combination of the 50Hz data message with the 1.023 MHz C/A code is transmitted using binary phase modulation on a sinusoidal 1575.42 MHz carrier frequency. By adding the 50 Hz data signal with the 1.023 MHz C/A code, the signal is spread over a wider bandwidth. In this way, the signal can be recovered, although the signal is transmitted roughly 20 DB below the thermal noise floor. To recover the signal, a code replica of the C/A code is correlated with the received signal. A correlation peak shows the presence of the C/A code in the received signal. A positive correlation peak indicates that the currently transmitted data bit is a '0', a negative correlation peak indicates a '1'. The military P(Y) code works in nearly the same way. Instead of the 1.023 MHz C/A code, a 10.23 MHz P code is used. This has the effect that the code is hidden deeper in the noise. To restrict access to this signal, the P code gets encrypted to the P(Y) code by adding a secret and very long pseudo-random sequence W to the P code. Both the length of the code and its hidden nature make it very hard to discover the P(Y) code. The C/A and P(Y) code are both transmitted on L1 with a frequency of 1575.42 MHz. The P(Y) code is shifted by 90 degree ($\pi/2$) in phase compared to the C/A code,(also called phase quadrature) so that the C/A code has its minimum and maximum when the P(Y) code is zero and vice versa.

Functionality of SAGA

We now describe how the military P(Y) signal can be used to securely authenticate the GPS signal, even without the knowledge of the secret W code. The main idea is to exploit the fact that the P(Y) code sequence received at location 1 is identical to the sequence received at location 2, except for the difference in satellite-to-receiver signal travel time (and some frequency differences due to the receiver clock and Doppler effect). The idea is to cross-correlate the samples taken at location 1 and location 2 with each other. This will result in a correlation peak when the phases of the P(Y) codes of the two samples are aligned with each other. The presence of this peak indicates the verifier that the same code is hidden in both samples. Of course, the C/A code in both samples can create a correlation peak as well. But keep in mind that the C/A and P(Y) code are orthogonal to each other. Hence, the verifier knows exactly where the P(Y) code should be located in the noise. The verifier takes sample points when the sine term of the C/A carrier goes to zero, so that the carrier of the P(Y) signal goes to one or minus one.[4] Hence, only the P(Y) code creates a correlation peak. The C/A code is known for each satellite. This enables the verifier to match each correlation peak to the different satellites, as the P(Y) code is in phase with the C/A code of the same satellite.

The verifier can precisely measure the time offsets between the satellite signals of the two samples to determine the arrival times of the different signals. If the verifier knows the arrival times of at least four satellites, the verifier can determine the exact location and time at which the sample was taken using the same trilateration methods that are used for the normal location determination in GPS. Hence, this system can be used to provide location verification from A to B. To provide location verification, the verifier B needs a signal S_B that is valid and has not been spoofed. When A sends a signal S_A to B, B can use the reference signal S_B and the signal S_A to verify the location and time at which S_A was received.

The accuracy of this methods lies within a magnitude of the normal GPS positioning determination. Hence, the verifier can determine the position where the signal was received with an accuracy in the meter range.

3 Threat Model Against Location Verification Using SAGA

In this section, the possible attacks on location verification using SAGA are described.

Signal-synthesis attacks using the secret code: The hidden signals are generated using a secret code. In the case of GPS, this code is the military P(Y) code. An attacker who possesses this code can generate valid signals for every

[4] In communication jargon, the verifier separates out the quadrature (sine) component from the in phase (cosine) component of the signal.

position he wants. Therefore, the system is only secure as long as the code is kept secret. If the attacker does not know the secret code used to generate the signals, he can try to guess the code. However, if the code is long enough and pseudo-randomly generated, it is computational infeasible for an attacker to successfully guess the secret code (which is true for the P(Y) code). It might be possible to use high-gain steerable antennas to raise the P(Y) code above noise. However, for every satellite one expensive and big (more than 10 meters) high-gain steerable dish antenna is needed. Note that if the P(Y) code is revealed, the anti-spoofing capability and the restricted use of the military GPS signals would be broken. Hence, the same security assumption is needed for the military GPS anti-spoofing mechanism.

Signal-synthesis attack without the secret code: An attacker can try generating and transmitting his own navigation signals. Such an attack is called signal-synthesis attack. However, the attacker does not know the secret P(Y) code and therefore the attacker's signals will not match with the verifier's signals. Hence, the attacker would need to attack the verifier's reference signal as well. An attacker can try a signal-synthesis attack by inserting a hidden signal h into the verifier's reference signal. To be successful, the attacker needs centimeter knowledge of the verifier's antenna. The attacker sends a hidden signal h to the verifier's antenna that is buried deep in the noise so that it does not interfere with the GPS signals. As the C/A code is not changed at all, and the hidden signals are buried well below the thermal noise floor, a verifier cannot detect the existence of these hidden signals. The attacker can now create a signal that the verifier falsely accepts as valid. To do this, the attacker first generates the C/A codes for the different satellites like they are expected at the wanted spoofing position. With the knowledge of the location of the verifier's antenna, the attacker can determine the travel time of the hidden signal to the verifier's antenna and therefore the offset between the hidden signal h and the C/A codes of each satellite in the verifier's data sample. Using the same offsets, the attacker aligns a copy of the hidden signal h with the C/A code of each satellite in his data sample. When the verifier correlates the data sample of the attacker with his data sample, the hidden signal h in the attackers's data sample and in the verifier's data sample correlates. The verifier cannot distinguish a correlation peak that is generated by the hidden signal h from a correlation peak that is generated by the P(Y) code and will therefore falsely accept the signal as valid. Without knowing the P(Y) code, the verifier will not be able to detect this attack. Signal observation techniques of the C/A code will be useless, as the original GPS signals are kept untouched by this attack. Using directional antennas to get the reference signal and shielding the antenna can make the attack much more complicated, as it would be more difficult for the attacker to insert the hidden signal into the verifier's signal. The insertion of the hidden signal can also be mitigated by collecting data samples from antennas at closely related locations (e.g. 3-5 meters). The attacker would need to align a hidden signal h_i for each used antenna i. Furthermore, each antenna would receive the

hidden signals h_i with a different phase. Using cross-correlation techniques the presence of these signals is detectable.The verifier can also use reference signals from different places to increase the security. By using signals from different locations a web-of-trust can be build. This would significantly increase the attack complexity as the attacker would need to spoof each of these locations. Note that signal-synthesis attacks become very complicated in case an attacker needs to attack a receiver over-the-air, e.g. when he attacks location self-verification. In this case the attacker needs to somehow get rid of the original C/A code and P(Y) code (if cross-correlation is used to detect spoofing), which can be very complicated without physical access to the user's receiver.

Delay attack: In a delay attack, the attacker delays the incoming signals for the same amount of time. If all signals are delayed for the same amount of time, this has no impact on the position computation. However, this results in a clock offset at the receiver. B can still validate at which time A was at the location L_A. But the clock of A and B are not synchronized. Therefore, delay attacks are very powerful against time synchronization, but do not have a direct impact on secure location verification. But if B's clock is not synchronized to a standard time reference such as UTC (from US Naval Observatory, GPS, etc.), e.g. because B is being attacked by a delay attack as well, B might falsely accept an old signal as fresh. Hence, as a requirement B's clock must be securely synchronized if B needs to decide whether the signal is fresh or not.

Selective-delay attack: In a selective delay attack, the attacker delays each satellite signal for a different amount of time so that a false position is calculated. This is a very powerful attack against navigation systems.[4] However, to be able to delay each signal for a different amount of time it must be possible to separate the signals from each other. But this is very difficult for the P(Y) signals as they are hidden in the noise. It might be possible to separate the signals by using high gain directional antennas for each satellite. Using a directional antenna pointing at one satellite, the C/A and P(Y) code of the target satellite are stronger than the signals from the other satellites.(But might be still below the thermal noise floor) If you combine signals from two directional antennas that target different satellites, a verifier might be able to detect the signals of the two satellites, while the signals of the other satellites might be too weak. Using this method the C/A and P(Y) signal from one satellite can be separated from the signals from the other satellites. But note that this attack needs at least four very good high-gain directional antennas and quite some knowledge in signal processing. Furthermore, this attack needs to be done in real-time, as the verifier can precisely determine the freshness of the signal. Whether this attack is successful depends on the verifier's ability to detect the signals from the not-targeted satellites. This strongly depends on the attacker's as well as the verifier's antennas and the verifier's effort to find these signals.

Relaying attack (wormhole attack): This is the most powerful attack against

location verification with SAGA. In a relaying attack, the attacker relays the signals S_v received at location L_v to the attacker's location L_A. As S_v is a valid signal, B will falsely validate A's position as L_v. This kind of attack is the biggest problem for all passive location verification techniques, as these techniques only verify the location of the received signal, and not of the receiver. So in passive location verification services that use GNSS techniques, it will only be possible to proof that an entity has access to a receiver (signal) at the claimed location, but not that he is actually there. As the exact reception time (less than a millisecond) of the signal is known these relay attacks can be made more difficult by setting up sharp bounds of the freshness of the signal. Note, that the accuracy of SAGA lies within the low meter range, hence, an attacker can only collect valid signals if he is within a few meters from the valid location.

Security Assumptions for Location Verification

We will now summarize the needed security assumptions in order to provide location verification services with SAGA.

1. *B can be sure that the signals he has received are valid and no other signal than the P(Y) code is hidden in the noise.*
2. *An attacker does not have a signal from the claimed location L_A for the claimed time period.*
3. *It is impossible to separate the signals from the different satellites from each other, so that they cannot be delayed for different amounts of time.*
4. *The attacker does not possess the secret code needed to generate the hidden signals.*
5. *Additional security assumption for location self-verification:* To prevent delay attacks, A either needs to be securely synchronized with GPS time or A needs to be sure that B's signal S_B is fresh.

4 Conclusion

The new mechanism to authenticate GPS signals is very promising to enable secure location verification services. As GNSS signals cover great areas, only about 6 reference stations can provide reference signals that enable location verification with global coverage. However, looking at the security assumptions it is clear that careful consideration is needed for every application to decide whether or not secure location verification is possible with SAGA. The key assumption for SAGA is assumption number 2, that an attacker does not have a signal from a valid location. This assumption is not just limited to SAGA but is rather a general shortcoming of location verification: If a malicious user has a collaborator at the claimed location, the verifier cannot distinguish whether the received location signals are the user's or the collaborator's signal. Hence, he will not be able to know whether the user is at the claimed location or some collaborator. Therefore, SAGA should be used in applications where it is very unlikely that

an attacker has access to a signal at a valid location at the claimed time. As an example application, a server with confidential information might restrict its access only to the company area and maybe the home of some employees that sometimes work from home. In this case, SAGA can be used for location based access control as an additional security mechanism for this server. Of course an attacker could try to circumvent this security mechanism by collecting a location signal at a valid location during the attack. However, this would significantly increase the complexity of the attack, especially as in many cases an attacker could be living far away from the target, e.g. in another country. Furthermore, the fact that the verifier will have meter knowledge of the attackers position, as well as the fact that he needs to be very close (in the meter range) to a specific location increases the chance that the attacker gets caught significantly. So location verification would not make the system unbreakable, but it could significantly increase the complexity of an attack. Hence, for many real-world systems, the proposed location verification techniques can significantly increase security. Furthermore, location verification can be a security tool that provides security in situation where traditional security mechanism such as passwords often fail. The main reason for this is that the security of location verification with SAGA does not rely on any secret information that can be lost and reused for later attacks. If a location signal is not fresh it is of no use for an attacker.

If SAGA needs to be resistant against very sophisticated attackers, the assumption that the P(Y) signals cannot be separated from each other might not be true, as an attacker could use very sophisticated high-gain steerable antennas. But in most cases, a possible adversary does not have access to such technology.

It should be further noted, that there currently does not exist any alternative to SAGA for using civil GNSS signals for location verification. Civil GNSS signals do not have any security mechanism so that spoofing can easily be done.[5] Hence, the security of SAGA far exceeds the security of normal civil GNSS services. Especially in applications, such as tracking, where location verification is the primary goal there is no alternative right now to SAGA with a comparable level of security when using GNSS. SAGA is also currently the most secure civil system for using GPS for location self-verification.

References

1. Denning, D.E., MacDoran, P.F.: Location-based authentication: grounding cyberspace for better security. Computer Fraud & Security (Feb 1996) 12–16
2. Lo, S.C., De Lorenzo, D.S., Enge, P.K., Akos, D., Bradley, P.: Signal authentication - a secure civil gnss for today. insideGNSS (Sep 2009) 30–39
3. Chandran, N., Goyal, V., Moriarty, R., Ostrovsky, R.: Position based cryptography. In: Proc. of CRYPTO 09. (2009) 391–407
4. Kuhn, M.G.: An Asymmetric Security Mechanism for Navigation Signals. In: Proceedings of the Information Hiding Workshop, Springer (2004) 239–252
5. Humphreys, T., Ledvina, B., Psiaki, M., O'Hanlon, B., Kintner, P.: Assessing the Spoofing Threat. GPS World (Jan 2009) 28–38
6. Scott, L.: Anti-Spoofing & Authenticated Signal Architectures for Civil Navigation Systems. In: Proc. ION GPS/GNSS. (2003) 1543–1552