Generating random numbers in hostile environments

Jan Krhovjak, Andriy Stetsko, and Vashek Matyas

Masaryk University, Faculty of Informatics
xkrhovj@fi.muni.cz, xstetsko@fi.muni.cz, matyas@fi.muni.cz

Abstract. This paper discusses basic security aspects of distributed random number generation in potentially hostile environments. The goal is to outline and discuss a distributed approach, which comes to question in the case of attacker being able to target one or several mobile devices. We define communication paths and attacker models instead of providing technical details of local generation. This paper also includes a discussion of several issues of such distributed approach.

1 Introduction

Since mobile devices typically use a wireless channel for communication, the security of transmitted data plays a very important role for many applications – consider, e.g., mobile banking. High-quality and unpredictable cryptographic keys, padding values, or per-message secrets are critical to securing communication by modern cryptographic techniques. Their generation thus requires a good generator of truly random and pseudorandom numbers.

The difference between truly random and pseudorandom numbers is given by the process of their generation. Truly random numbers are typically obtained by sampling of some physical phenomenon (e.g., thermal noise) while pseudorandom numbers are computed by a faster deterministic algorithm. The classical cryptographic pseudorandom number generators use truly random data only as an initial input (seed) to the algorithm.

The security of local generation of truly random (and also pseudorandom) data relies primarily on the quality of used sources of randomness. The mobile phones typically provide some good sources of randomness – e.g., noise present in audio and video input – that we analyzed in [KSM07]. However, the possibility to predict and/or influence such sources of randomness implies a possibility to predict generated data.

The security of pseudorandom number generation relies on the design of particular cryptographic pseudorandom number generator and its resistance to cryptanalysis. In addition to that, modern *hybrid generators* periodically use truly random data during the whole generation process – this improves the generator security by increasing resistance against state compromise attacks at the expense of higher demands on truly random data.

Since mobile devices or their sources of randomness can be under attack – consider, e.g., malware or influencing video input by changing ambient light intensity – we can involve several cooperating mobile devices in the generation process. These devices can perform generation at the beginning of (or during ongoing) communication with other devices. This distributed approach can support better random or pseudorandom number generation in case of attackers being able to target only some (but not all) of the mobile devices.

Local generation, from the attacker point of view, is obviously strongly dependent on the attacker possibilities to control a mobile device and to influence or predict used sources of randomness. The situation is quite different when we consider distributed random data generation. In this case, the attacker possibilities depend also on the communication model and methods for secure gathering and using remotely generated data.

The rest of this paper is organized as follows: In the next section, we define attacker models for local random number generation. Section 3 focuses on definition of basic communication paths and describes several problems that we encountered. Section 4 sketches possible mechanisms for gathering random data in hostile environments and discusses problems that should be considered.

2 Attacker model for standalone mobile devices

Recall that random data for cryptography purposes must have good statistical properties and must be unpredictable. These two conditions are jointly satisfied only if the truly random data are generated with utilization of a good physical source of randomness and postprocessed by a cryptographic pseudorandom number generator.

A successful attacking (i.e., unobservable influencing) of such sources can result in non-uniform random data or in completely predictable (or even worse constant) data that is not random. Utilizing several sources of randomness and combining their outputs is a common practice to avoid a prediction of a generator output in the case when the attacker influences some (but not all) sources of randomness. Better statistical quality and faster generation (without increasing of entropy) is accomplished by utilizing digital post-processing, e.g., by cryptographic pseudorandom number generator. Hybrid generators then also allows to increase the inner state entropy by periodical reseeding and continual accumulation (pooling) of truly random data.

We often assume that the attacker has no access to the generating device – in this case the postprocessing can hide many statistical defects or even influence the source of randomness that results in generation of constant data without entropy. The situation is more difficult if an attacker somehow obtains an inner state of pseudorandom number generator – e.g., due design flaw, implementation error, or by readout of the memory content. In this case, the attacker can also easily predict all pseudorandom data before next reseeding. Potential simultaneous influence of the randomness source then allows to predict pseudorandom outputs even after reseeding.

Currently, all mobile phones that want to access mobile network are equipped by the subscriber identity module (SIM). It is a smartcard that provides secure storage, secure computational environment, and it also contains physical truly random generator – typically based on sampling of several free running oscillators. However, SIM cards are under control of the mobile network operator and there are very limited possibilities of their usage by common users – often restricted only to the secure storage of contacts or short text messages (SMS).

The future technical progress may result in mobile phones with second smartcard that will be under full user control. In this case, all cryptographical operations, including generation of random numbers, could be performed inside the card similarly as in classical SIM Toolkit applications, and the external sources of randomness can only serve as an additional (but non-reliable) input.

The main problem of mobile devices is that their computational environment is not secure. Such devices can contain malware (malicious software as viruses, Trojan horses, etc.) and all generated random data could be easily replaced by non-random data before they reach the appropriate application (located in mobile device or inside SIM card). One possibility to prevent unwanted malicious software or even firmware installation lies in the introduction of a trusted platform module (TPM).

All this implies that the real attacker model for standalone mobile device is strongly dependent on the attacker possibilities to control the device or used sources of randomness. We define four classes of attacker for standalone mobile devices:

Type I (weak outsider) – the attacker had temporary read access to the mobile device and knows the internal state of device – including pool – before beginning of the generation. He has no possibility to access to the mobile device again, but he has access to the information about the environment of victim (he can stay in the proximity of victim, he can record audio/video of the victim to the camera, etc.) and he has also a limited capability to influence this environment (e.g., disturbing the signal, overexposure the lens of digital camera, etc.).

Type II (strong outsider) – the attacker has in addition to the weak outsider almost all detail information about victims environment and he is capable fully influencing this environment. The term "almost" reflects uncertainty arising from interactions between user, device, and environment (several physical effects, errors in measurement, etc.).

Type III (weak insider) – extends the capabilities of previous strong outsider by adding a full control over the mobile network operator SIM card with the possibility of remote reading from or writing to the SIM card.

Type IV (strong insider) – in addition to previous scenario, the attacker has a temporal write access to the mobile device. Therefore, he can also compromise the firmware/software of the mobile phone (e.g., by rewriting flash, installing malware) and hence he has a full control over the phone (including interprocess communication) with a possibility to remote access to the mobile device again.

Clearly, a digital postprocessing by cryptographic pseudorandom number generator causes that an attacker always needs to know the device internal state. Hoverer, as we discussed in the introduction, a careful user should always expect that sources of randomness in a standalone generating devices can be under attack. A successful attack (performed by weak or strong outsider) and the knowledge of internal state always results in a predictable data output.

It is extremely difficult to guarantee that the external source of randomness is not under an ongoing attack. Several online statistical tests can be performed, but they can probabilistically detect only a basic (and limited) set of statistical defects. An additional and more convenient way how to secure a pseudorandom output is to prevent attacker from copying the internal state by utilizing secure storage inside the SIM card. This works only until the mobile network operator starts to act as/with an attacker, being capable to read/write content of SIM card that is in his ownership.

Preventing attacker with full access to the device is the most difficult task that can be meaningfully accomplished only by introducing a trusted platform module. Since we want to keep our discussions realistic, we expect a type four attacker being able to target several (but not all) remote devices. However, we assume the local devices (including SIM card) behave always correctly. This guarantees (in terms of probability) that a trustworthy local device obtains at least some random data from remote parties and thus is resistant against first three types of attackers. The clarification of this strict assumption is described below.

3 Communication model

Since we are interested in random data generation in mobile environments, we will distinguish between *consuming mobile device* that requests random data and *generating mobile device* that (e.g., upon a request) generates random data. Sometimes we consider a *generation computer* located in the Internet or GSM network that (e.g., upon a request) also generates random data.

In the basic scenario we expect that the owner of a trustworthy consuming device always selects trusted remote users to generate random data. Particular generating device replies with a message that includes requested random data and declaration about the amount of entropy in this data. Based on the user reputation the consuming device makes a decision about the amount of claimed entropy of the obtained random data. This reputation can be predefined by the consuming device owner and we call it static reputation.

Unfortunately, there is no way how to assess the statistical quality of obtained sample of random data and how to validate the amount of entropy in such data. Even a device of a user with good reputation could become the victim of the malicious code (viruses, Trojan horse, etc.) that can produce only pseudorandom data with no entropy. This prevents also using all dynamic reputation system that automatically recalculates reputation. The only meaningful solution of this

problem is using random data from several devices where at least one device is expected to be honest and the communication between devices is well secured.

In this section we focus on the communication issues and we restrict ourselves to the situation when both communicating devices behaves correctly.

3.1 Communication paths

For a precise definition of attacker models in the distributed environment, it is essential to know the communication model that includes network topology, used security mechanisms and their fundamental vulnerabilities. All these communication properties are briefly discussed in Appendix A and a detailed description of these issues can be found, e.g., in [EVB01] or [Xia07].

We define several path types that are used in definitions of attacker models. The device at the beginning of path is always a consuming mobile device; the device at the end of the path is the generating mobile device or computer. The path can also lead through the Internet and the first computer that provides Internet connection to that devices on end-points of the path is denoted as a *gateway*.

Type 1 – the simplest local path can be established between two mobile devices or a between mobile device and a computer. These paths can be point-to-point (via, e.g., IR or USB interface) or point-to-multipoint (via, e.g., Bluetooth or WiFi). Paths between two mobile devices can lead over one or several intermediate devices. For example, in the case of Bluetooth the path can lead over one superior/master device. Another example is a large WiFi network where two mobile devices can be connected to different access points. Therefore, communication path between these mobile devices can lead through several access points.

Type 2 – the GSM communication path established between two mobile devices can lead over several GSM networks. These paths can be created by standard GSM technologies (e.g., SMS or MMS). Moreover, the mobile network operator has a capability to improve his network by additional special GSM services that can extend network functionality (e.g., servers that provide on demand random data). In this case the communication is established between mobile device and such GSM service server.

Type 3 – mobile device can communicate with other mobile devices or computers through the Internet. The access to the Internet can be established through a gateway, which could be personal computer or wireless access point. The path between consuming mobile device and gateway (and between gateway and generating mobile device) is covered by the simplest paths of "type one". Internet or similar packet oriented network (based on TCP/IP, X.25, etc.) is either used between gateways or between a gateway and the generation computer.

Type 4 – hybrid paths through the Internet where one or two gateways on the path are in fact GPRS support nodes of different GSM networks. The path between consuming mobile device and gateway (and between gateway and

generating mobile device) is covered by the paths of "type one" and "type two". For example, consumer mobile device can request random data from the generation computer or the generating mobile device connected to the Internet through the gateway – another computer or access point. Moreover, the gateway for the generating mobile device can also be a GPRS support node.

Note that leased lines can be used to interconnect different GSM networks. The description of mechanisms that secure data flow in such lines is not publicly available. The lack of this information implies that the leased lines should not be trusted even when the mobile network operator (or its employees) behaves honestly.

3.2 Attacker model for distributed systems

Since the attacker is able to eavesdrop some communication links, they have to be secured in terms of authenticity, confidentiality and integrity. In order to design secure systems, which support distributed random number generation, we should take into consideration the communication paths and the corresponding attacker models. (We enclose the attacker models in Appendix B.)

We define four different attacker types according to the communication paths described above. However, as we are not able to detect potential modification or observation of the transferred data, we assume that each particular communication link in the path must be secured. This can be done either by the owner of the infrastructure (e.g., GSM network operator) or by end-point mobile device (by means of end-to-end security). This implies that our attacker models degrade and all types of attackers can be prevented by securing the whole communication path.

In the next section, we consider the cryptographic protocols that allow two or more distributed parties to establish a shared secret key and contribute to the process of its generation. The authenticated protocols are designed to work in hostile environment and to provide end-to-end security, and can be used to prevent all attacker types.

4 Gathering of random data in hostile environments

As was mentioned above, the local random data generation can be performed in a hostile environment and the mobile devices and their sources of randomness can be under ongoing attacks. Therefore, we suggest to involve more parties (mobile devices) in the process of random data generation. Such mobile devices can be considered as independent (remote) sources of randomness[1]. In general, the more generating mobile devices are used, the less probability to attack all of them is – due to usage of different devices, environment, and to certain point also a communication paths.

[1] The remote mobile device can provide both raw and postprocessed random data.

Consuming device can obtain random data from generating devices either per explicit request or as a secondary product of ongoing communication (e.g., audio/video conference). In the former case the user sends the request for random data to one or several generating mobile devices. In the latter case the random data are transferred during communication. This functionality can be supported either by a mobile network operator service or by a third party application, which in turn uses existing network services (e.g., SMS, MMS).

In the distributed environment we can also distinguish between two methods of obtaining the random data per explicit request – direct or indirect. In both methods the consuming mobile device requests another device to provide random data. Direct method means that the response is sent directly back to the consuming device. Indirect method, on the other hand, means that the response is sent through another mobile devices (can be predefined by user), which add their own random data. The last device sends the accumulated random data back to the consuming device. It is an open question whether such method brings some significant advantages (e.g., for ad-hoc networks) and so would be more effective than the direct one.

Technology improvement (e.g, 3G/4G networks) introduces the possibility of audio/video conferences. To assure confidentiality in such conferences, the participants typically have to agree upon a shared secret key, which is used to encrypt the ongoing communication. This scenario requires all participants to contribute the random data to the shared key. This fact benefits particularly the consuming devices which are locally influenced by the attacker (type one or two). The shared key could be treated as a random data generated in the distribution manner.

The disadvantage of this method is that all participants share the same random data that are predictable (with no entropy) for an adversary inside the group. Therefore, we propose to keep a distinct pool of random data for each communication group. The random data stored in a pool associated with a articular group can be used to secure communication only within this group.

4.1 Distributed contribution protocols

In this section, we discuss (multi-party) cryptographic protocols that can be used for distributed random data generation. These protocols enable each user to accumulate random data from multiple participants – therefore, we call them *distributed contribution protocols.*

We consider the group key agreement (GKA) protocols as possible candidates that can be utilized for distributed generation of random data. A brief description of several basic GKA protocols can be found in [BM03]. These protocols are typically based on Diffie-Hellman key establishment and work in several rounds. Each participant of protocol generates new (or modifies obtained) Diffie-Hellman value(s) and sends it/them back to the initiator (or to the next member) of the group. The detailed messaging is dependent on particular logical topology of a protocol, however the modification of exchanged values always involves the usage of randomly generated data.

These protocols provide either no authentication or many-to-many/many-to-one authentication (typically based on secret key/password of remote party). The many-to-one authentication scheme assumes that all participants authenticate themselves only to one participant (e.g., initiator of the communication) and they have no direct assurance who are the group members. The non-authenticated protocol could be easily transformed to the authenticated one – for example, by the means of digital signatures.

More sophisticated versions of protocol are designed to fit to concrete physical topology (e.g., GSM network), but the advantage here is only in more effective messaging or in offloading of complex computations from resources-restricted mobile devices [BCE04]. Another class – fault-tolerant GKAs – allows to detect parties that do not follow the protocol [Tze05]. However, since the underlying Diffie-Hellman problem is considered as hard, all exchanged values can be always without any harm observed by an attacker.

Note that classical authentication protocols often rely on shared secrets that are typically stored inside the device and this fact implies only the device authentication. More sophisticated password-based authentication protocols (e.g., password-based GKA) provide better user authentication, but also often requires more random data. Several innovative methods allows perform user post-authentication by audio-visual means, which requires even less random data then classical authentication. This kind of authentication relies on the ability to recognize the user face/voice and other behavioral characteristics [LP08]. Another scenario utilize visual checking of exchanged Diffie-Hellman values that can be for easier verification transformed to usual language words [CH04].

4.2 Chicken-and-egg problem

The distributed approach to random data generation has one significant drawback. As we mentioned above, the transfer of random numbers must be secured. However, common mechanisms for ensuring authenticity, confidentiality and integrity (but also, e.g., anonymity or information hiding) are based on classical cryptography, which in turn is dependent on random data as secret keys, padding values, etc. This implies a classical chicken-and-egg problem and breaking this circle seems to be impossible – from both information-theoretic and complexity-theoretic points of view.

The main reason is that encryption of high-entropy data behaves similarly as a pseudorandom number generator. The maximal entropy that can be obtained from encrypted data is limited by the entropy of the secret encryption key. We cannot count on more entropy then the entropy of the encryption key, because the adversary has a possibility to attack the secret key. The solution could be reestablishing a shared key for each request for random data, however, this process requires that all involved parties have a reliable source of random data.

Despite this drawback, we propose to use both raw and postprocessed random data obtained from remote parties as an independent additional input for next digital postprocessing – e.g, by a hybrid pseudorandom number generator or a

nondeterministic entropy extractor. Such secure nondeterministic transformation is the only way how to solve the problem that the whole group shares the same random data, usage of which then has to be restricted to this group. We are aware that both techniques require some amount of truly random data and the entropy outside the group will be restricted to the entropy of that truly random data. The main benefit here is that we obtain more secure and reliable method of generation (regardless of entropy) in the presence of attacker type one or two.

As we have mentioned, the direct usage of random data obtained from remote parties can be done only for securing communication within a particular group. Due to the entropy issues, we recommend its usage only for short-term encryption keys or other secrets that are intended for applications where the data to be protected are sensitive only for limited time period – e.g., daily stock price forecasting or common day-to-day audio/video conferences. Other direct usage without digital postprocessing is not recommended. Note that this holds also for local generation in the presence of attacker type one or two, especially in the case when we want use local random data for first key agreement or establishment.

The only way to completely avoid usage of digital postprocessing is utilizing the SIM card at least to establish a secure communication channel. In this case is also recommended to use the secure storage of the SIM card to protect seed files and all randomness pools.

5 Conclusion

Since a mobile device (or its sources of randomness) could be influenced by an attacker, there will be scenarios where one should consider extending a local random number generation to the distributed one, with several mobile devices involved.

We examined various communication paths that can be used to interconnect mobile devices (e.g., Bluetooth, GSM, Internet, etc.). Any proposed secure system has to be designed under a defined attacker model. Therefore, we looked closely at different attacker types according to the communication paths involved. Further, we discussed multi-party cryptographic protocols that could be used for distributed random data generation. We considered one class in more detail – the group key agreement protocols, where the shared secret key could be treated as random data generated in a distributed manner.

Our aim was to point out distributed random number generation as a possible way to obtain high quality random data. Obviously, even the distributed approach has its advantages and disadvantages. We presented some possible solutions and we hope to encourage a futher discussion of this approach.

References

[KSM07] J. Krhovják, P. Švenda, V. Matyáš. The Sources of Randomness in Mobile Devices. In Proceeding of the *12th Nordic Workshop on Secure IT System*, pp. 73–84. Reykjavik University, 2007. ISBN 978-9979948346.

[EVB01] J. Eberspaecher, H.-J. Voegel, C. Bettstetter. *GSM Switching, Services, and Protocols*. Wiley, 2001. ISBN: 978-0471499039.

[Xia07] Y. Xiao. *Link Layer Security in Wireless LANs, Wireless PANs, and Wireless MANs*. Springer, 2007. ISBN: 978-0387263274.

[BM03] C. Boyd, A. Mathuria. Protocols for Authentication and Key Establishment. Springer-Verlag, 2003. ISBN: 978-3540431077.

[LP08] S. Laur and S. Pasini. SAS-Based Group Authentication and Key Agreement Protocols. In Public Key Cryptography (PKC) 2008, LNCS 4939, pp. 197–213, 2008.

[CH04] M. Čagalj and J.-P. Hubau. Key agreement over a radio link. *EPFL-IC Technical Report*, No. IC/2004/16, 2004.

[Tze05] Y.-M. Tseng. An Improved Conference-Key Agreement Protocol with Forward Secrecy. In *Informatica*, Vol. 16, No. 2, 275–284, 2005.

[BCE04] E. Bresson, O. Chevassut, A. Essiari, and D. Pointcheval. Mutual Authentication and Group Key Agreement for Low-Power Mobile Devices. The *Fifth IFIP–TC6 International Conference on Mobile and Wireless Communications Networks*, pages 59–62, 2004.

Appendix A – Basic communication properties

In the following text we briefly summarize basic information about GSM network architecture and communication techniques of mobile devices.

GSM network architecture

The GSM (Global System for Mobile Communication) network topology typically consists from two or three main subsystems. The first is *base station subsystem* that consists of mobile phones, base transceiver stations (BTS), and base station controller (BSC). Mobile phones are in fact generic devices that are personalized by subscriber identification module (SIM) card. BTSs are responsible for wireless connection with mobile phones and BSCs manages multiple BTSs and performs spectrum allocation as well as handoffs.

The second is *network and switching subsystem* that consists of mobile switching center (MSC) and several registers and databases. MSC manages multiple BSCs and provides connection to wired telephony network. Registers and databases are necessary for authentication, network management, storing information about (roaming) subscribers, etc.

The last subsystem called *GPRS core network* is used to for support general packet radio services (GPRS). It consists of several centralized GPRS support

nodes (GSN) that allows connection and data transmissions to Internet. The most important GSNs are serving GPRS support node (SGSN) that is responsible for delivering data packets, and gateway GPRS support node (GGSN) that is gateway between GPRS and external packet data networks.

GSM security mechanisms are authentication (challenge-response protocol), confidentiality (encryption by symmetric cipher), and anonymity (temporary identities). However, only the user must be authenticated to the network and confidentiality of data (voice, signaling information, etc.) is preserved only on the radio path. The exception is GPRS core network because all data transmissions over GPRS are encrypted on the path between mobile phone and SGSN.

These security features imply obvious properties and vulnerabilities of GSM networks – e.g., using false BTS station that can force mobile phone to disable encryption, interception of communication on remaining part of the network, lawful interception, etc.

Communication techniques of mobile devices

In the previous section we restricted ourselves to discussion of GSM/GPRS architecture and its security issues. We shall consider all mobile devices capable to connect GSM networks (including PDA phones, notebooks with GSM PCMCIA cards, etc.) because many of them have own O/S and support various applications that may require well-secured communication. We shall also summarize and keep on mind all possible methods of communication that can be established by these devices and thus can be a subject of various attacks. Note that we use GSM/GPRS as our reference mobile network, but similar discussion can be done, e.g., for UMTS/HSDPA or other 3G/4G networks.

The most versatile mobile devices are, from our point of view, PDA phones. These devices have own operating system, can run various applications, and support a lot of communication techniques:

1. Infrared (IR) – short-range wireless link that requires direct visibility of communicating parties. IR link is unprotected and uses no authentication or encryption.
2. Bluetooth (BT) – short-range wireless radio link. Direct visibility is not required and thus password-based authentication techniques are used. First successful authentication (so-called BT pairing) allows paired devices to communicate without further authentications. The communication is protected by stream cipher E0.
3. Wireless access point (WiFi AP) – medium-range wireless radio link that supports optional unilateral user authentication and encryption (e.g., WEP or WPA).
4. GSM/GPRS network – radio link between mobile device and BTS is protected by A5 algorithm but particular mobile operator knows all secret keys in his network. Transmissions in wired part of network (with exception of GPRS) are typically unencrypted.

5. USB interface – unprotected wired link typically between mobile device and computer that can share connection to the Internet.

The communication between mobile devices or between mobile device and some (semi)trusted server always utilizes some of these techniques.

Appendix B – Attacker model for distributed systems

The attacker models are divided into the four different types according to the communication paths described above. Note that we always expect that consuming device is trusted and cannot contain malicious software as viruses, Trojan horses, etc.

Attacker type 1 can exploit weak points of communication path type 1.

The USB and IR point-to-point communication link use neither authentication nor encryption mechanisms. That fact implies that an attacker can eavesdrop all the communication and impersonate any device. For example, the attacker taps the USB cable. However, such attacks are not easy to be done due to the physical characteristics of the USB and IR links. Both of them provide short-distance communication, but USB cable is under control of its owner and IR requires direct visibility between communication devices.

The WiFi and Bluetooth communication link use authentication and encryption mechanisms. WiFi authentication is unilateral – only end-point devices have to authenticate themselves to access point (in infrastructure mode) but not vice-versa. That leads to the fact that an adversary can act as legitimate access point. Bluetooth authentication is based on the knowledge of the shared password and is bilateral. In case of three or more parties involved into communication path Bluetooth and WiFi have the same security flaw. Man-in-the-middle attack is always possible, because secure protocols are always performed only between the master/AP and slave/end-point devices and never between two slave/end-point devices.

The link could be secured using encryption. However, Bluetooth encryption algorithm E0 is considered to be weak. WEP used for securing WiFi has problems with key management and initialization vectors setup. Several access points can be connected by either wired or wireless links that also could be attacked by an adversary.

Attacker type 2 can exploit weak points of communication path type 2.

GSM network requires mobile device authentication (algorithm A3/A8 also called COMP128), but GSM network does not authenticate itself to the mobile device. That fact gives the attacker the possibility to create fake BTS, which acts as an

original one. Such attacker can perform man-in-the-middle attacks or can disable A5 encryption between mobile devices and BTS. Moreover, A5 algorithm is not public and some previous versions were broken.

A more disturbing attack scenario involves untrusted mobile network operator (including its employees) or/and lawful interception performed by the government.

Attacker type 3 can exploit weak points of communication path type 3.

Communication path type 3 extends communication path type 1, therefore, it contains all path 1 weaknesses. Since the communication path type 3 goes trough the Internet or similar packet oriented network it also suffer from vulnerabilities typical for this type of networks.

Attacker type 4 can exploit weak points of communication path type 4.

Communication path type 4 extends communication path type 3, therefore, it contains all path 3 weaknesses. Since the communication path type 4 goes trough the GSM network it also suffer from vulnerabilities typical for that type of network.