Reflections on Evolving Large-Scale Security Architectures

Geir M. Køien

Institute of ICT
Faculty of Engineering and Science
University of Agder, Norway
Email: geir.koien@uia.no

Abstract—**In this paper, we conduct an informal analysis of evolving large-scale security architectures. The 3rd generation partner project (3GPP) mobile systems is our example case and we shall investigate how these systems have evolved and how the security architecture has evolved with the system(s). The 3GPP systems not only represent a truly long-lived system family, but are also a massively successful system family. What once was an auxiliary voice-based infrastructure has evolved to become a main, and thereby critical, information and communications technology (ICT) infrastructure for billions of people. The 25+ years of system evolution has not all been a linearly planned progression and the overall system is clearly also a product of its history. The goal of this paper is to capture some of the essence of security architecture evolution for critical ICT systems. What makes the evolution work and what may break it? These are important issues to analyse, and this paper aim at highlighting some of the aspects that play a role in security architecture evolution. In this sense, the paper is about research directions.**

Keywords–Evolving Security; System Security; Security Architecture; Long-term security planning; Migration; Mitigation; Deprecation.

I. Introduction

This paper is an extended version of the paper "Challenges for Evolving Large-Scale Security Architectures" [1] (SecurWare 2014). The scope has been broadened and significant extensions has been made. In particular, we have added new material to Sections III and V, and Section VI is entirely new. Other amendments have been made throughout the paper.

A. Background and Motivation

The example system investigated in our study, the 3GPP systems, has gradually become important, all-encompassing and pervasive on a global scale. Initially, the systems only served as an auxiliary and adjunct infrastructure, but gradually it has replaced the fixed line telephone. Today, the 3GPP mobile system services are pervasive and ubiquitous, and the systems have also become a major IP infrastructure. The convenience of mobility has been a major driver, and now the mobile infrastructures are poised to become the major access network for the fast growing Internet-of-Things (IoT) and the machine-to-machine (m2m) ecosystems.

Security was never a big priority in the 1G analog cellular systems. Originally, there were not any security problems either, but with opportunity and almost non-existent protection came theft and fraud. So, when the 2G digital systems came, the need for security was recognized and one devised mechanisms to address the threats from the first generation [2].

When the 3G architecture was designed, there were no serious practical problems with 2G security as such. However, it was clear that the authentication was too weak, that 64-bit encryption was not going to be enough and that the scope would not suffice for IP connectivity [3]. The 3G security model therefore improved on existing schemes to address the known shortcomings [4]. Additionally, one added support for core network protection (profiled use of IPsec) [5]. With the advent of 4G security one found that weaknesses induced by backwards compatibility with 2G was a most urgent problem to be fixed. Of course, one also took the opportunity to fix some of the other shortcomings of 3G security. To this end, the key-deriving key hierarchy in 4G was a clear improvement, both security-wise and performance-wise [6]. During the progression from 3G to 4G a lot of core network protection measures were added, but these were not so much part of a design as a patchwork of useful, but specific schemes. In the meantime, the importance of the systems have far outgrown the added protection capabilities, and as technology and scope has progressed it is clear that the 3GPP security architecture has not really evolved far enough to cater for the new needs.

The 3GPP security has rightly been criticised in academic circles. However, much of the criticism is somewhat misguided. That is, the criticism may technically be accurate. However, the impact of a theoretical cryptological weakness are often negligible in practice, and the suggested improvements are often utterly impractical to deploy in an existing system. To neglect to cater for migration is a major showstopper in practical terms. Obviously, there is also organizational politics at play, which makes it even harder to make changes, particularly if the benefits are perceived to be minor. The proverb "*If it ain't broke, don't fix it*" comes to mind.

To criticize the 3GPP security architecture is one thing, but it is rather more interesting to try to investigate the ways forward. In particular, how does a security architecture evolution actually work. What can we learn from the 3GPP case and how can we use this to make it work better. The goal is not completeness or a full understanding, but rather to identify key aspects that define evolving security architectures.

Security is difficult and hard to get right. Good cryptographic primitives are very hard to design and it is even harder to verify that they do not have any fatal flaws. Still, by and large, this is doable. Cryptographic protocols are also very hard to get right. We know quite a lot about how to construct communication protocols, and there are many formal verification methods that allow us check for a whole range of properties. Still, it is very hard to design a secure cryptographic protocol. There are tools that will allow us to check security protocols (see [7, 8]) for certain security properties, but overall the state of the art for security protocol design is immature [9].

When it comes to security architectures, we are often dealing with complex systems that needs to be secured. The problem is highly complex and it must be dealt with on many different levels. This problem is not well understood, and yet it is vital that these complex system will be properly protected. This situation the motivation for this paper. It must be seen as an initial effort. We hope to improve on situation awareness and we hope to inspire more future work in how security architectures evolve. This, we believe, will provide us with tools to make better, more resilient and robust systems.

B. The 3GPP System Context

Mobile radio existed before we got fully automated systems. Systems like the analog 1G Nordic Mobile Telephony system, which had unassisted call setup and automatic handovers, marks the start of true cellular systems around 1980. The first 3GPP system is the second generation (2G) Global System for Mobile communications (GSM), developed in the mid/late 1980ies. Originally, GSM only featured circuit-switched (CS) services, but was later adapted to also include packet-switched (PS) services through the General Packet Radio Service (GPRS) extension. With the new millennium came the third generation (3G) Universal Mobile Telecommunications System (UMTS), which natively features both CS and PS services. From around 2010 we also have the fourth generation (4G) Long-Term Evolution (LTE) system, which is a broadband PS-only system. LTE is further developed into LTE-Advanced (LTE-A).

1) Principal Parties: From a subscriber perspective, the system can be described with three types of principal parties.

- The Home Public Land Mobile Network (HPLMN).
- The Visited Public Land Mobile Network (VPLMN).
- The subscriber/user (USER).

These parties also represent legal entities, and the relationships are determined by contractual agreements. It is immediately clear that while the number of HPLMN and VPLMN operators will be limited to a few thousand, the number of subscribers will easily be in the billions. There is also a distinction between a subscription and a legal entity in that a person or organization may own many subscriptions, and this will certainly be the case for IoT/m2m subscriptions.

A national telecom regulator will also be involved, in addition to external service providers. Over-national regulatory bodies also exists, but their influence will likely be mediated by the national regulator. One may also add intruders to the list. The external service providers usually have little influence on how the networks operate and so we exclude those for further discussion. Likewise, in this context, we do not see a need for including virtual mobile network operators (VMNOs).

2) System Development: The 3GPP system specifications are developed by the 3GPP, but ratification is done by the organizational partners (formal standardization bodies). The design is "*contribution-driven design-by-committee*", and the process is largely consensus driven. The contribution-driven aspect quite literally means that company impact is relative to the number of contributions. Normally, it will be enough if 4 companies sign up for commitment to develop a feature. By-and-large, there is no real way to stop initiatives, and so the architecture sometimes suffer from new developments that do not really fit well with the overall architecture. Initiatives to develop new features may of course be stopped, but this is more likely to be caused by patent issues etc. than related to system architectural concerns.

The impact is noticeable when it comes to priorities and efforts spent. Early on, when GSM/GPRS was specified, the operators took considerable responsibility and led many of the efforts. Subsequently, the vendors have taken over more and more of this work. The impetus to carry out work is clearly related to the business potential the work has. Unfortunately, investments in security functions seldom look like a good business proposition prior to an incident.

3) Mandatory Features: The 3GPP differentiates between *mandatory for implementation* and *mandatory for use*. That is, a feature may be mandatory to be implemented by the vendors if they want compliance with a system release. At the same time, the operators may freely disregard the feature if they want. Other functions may be mandatory both to develop and deploy. In terms of deployment, this often means that the features that are not mandatory for deployment will only get deployed at a later stage, if at all.

4) Scope: The 3GPP scope has extended over the years and so has the scope of the security protection. However, aspects such as server hardening and similar is still considered well outside the scope, and generally one limits the scope to the protocols directly developed by 3GPP or for features that are otherwise captured by 3GPP specifications. Except for where interoperability is at stake, one generally avoids schemes being *mandatory for use*.

5) Licenses and Regulatory Requirements: Cellular systems operate in licensed bands and are subject to regulatory requirements. These requirements include support for lawful interception (LI) and emergency call (EC) [10, 11]. The last decade we have also had anti-terrorist measures such the EU Data Retention Directive (DRD) [12].

C. Brief Introduction to 3GPP Systems

1) 2G – GSM and GPRS: The GSM and GPRS systems are the 2G systems. It is common to see monikers like 2.5G used for GPRS, and 2.9G used for GPRS with Enhanced Data rates for GSM Evolution (EDGE). The main GSM features are mobility, speech and text messaging. GPRS is an overlay system to GSM. It features two additional core network nodes and provides PS support. With EDGE (new codecs) it provides up to 236 kbps data-rate. There is also an "Evolved EDGE" extension on the horizon, with yet higher data-rates. The 2G-based radio access network is called GSM EDGE Radio Access Network (GERAN).

2) 3G – UMTS (incl. High-Speed Packet Access (HSPA)): The UMTS system was finalized in late 1999 and is a combined CS/PS system. It can readily achieve >10 Mbps data-rates (w/max. rates >100 Mbps downlink). The system is a mix of GSM/GPRS technology and protocols and, increasingly, IP-based protocols and technology. The radio access network is called the Universal Terrestrial Radio Access Network (UTRAN).

3) 4G – LTE and LTE-A: The LTE systems are designed as all-IP networks (AIPN) and features true mobile broadband. The core network is fully IP based and there are no CS components in LTE. The radio system is highly advanced and provides true broadband services. The radio base-stations, called eNB, are logically mesh connected. There are no longer any controllers in the access network (E-UTRAN). The VPLMN mobility functions are carried out by the mobility management entity (MME) server.

D. Paper Layout

In Section II, we briefly outline the security of the 3GPP systems. In Section III, with investigating what evolution means in the context of a security architecture. Then we proceed in Section IV, were we discuss what may induce changes in a security architecture. This is followed up in Section V with some assumptions regarding the security architecture and the system context. In Section VI, we take a look at some of the factors that come into play and that might cause problems and even outright failure for a security architecture evolution. These factors are almost exclusively non-technical ones. In Section VII, we try to learn from the lessons and provide some advice. In Section VIII, we try to distil actionable knowledge from the previous sections. Finally, we sum up our effort and provide some concluding remarks in Section IX.

II. Security in the 3GPP Systems

In this section, we provide a short description of the main features of the 3GPP security provisions.

A. 2G Security

There is no well-defined security architecture per se in the 2G systems. The main security specification was technical specification (TS) 03.20 “Security-related network functions”, which subsequently has been transposed into TS 43.020 [2]. It defines the identity- and location privacy scheme, the entity authentication protocol and the smart-card based security functions. It also outlines the over-the-air cipher function(s). The over-the-air ciphers must be supported both by all access networks and user equipment (UE). These ciphers must therefore be fully standardized. Figure 1 outlines the GSM security procedures. The scenario consists of the user equipment, the visited network and the home network.

1) Background and Requirements: In the voice-only 1G systems one had experienced charging fraud and impersonation fraud. Two distinct types of attacks quickly came into focus:

a) Eavesdropping was a problem as the analogue voice channel was unprotected and easy to listen-in on.

b) Faking the call setup signaling, which was digital, was quite easy and could in principle be done by simply recording a setup sequence and then later replay it.

A main priority for the second generation system GSM was therefore to **a)** protect the over-the-air channel against eavesdropping, such that it would no longer be the weakest link, and **b)** provide credible subscriber authentication to avoid impersonation attacks. The fact that GSM featured digitally encoded speech made protection much easier, as it permitted use of encryption.

2) The 2G Security Architecture: GSM security is based on a physical subscriber identity module (SIM). For portability reasons it was decided to use a smart-card. The SIM comprises both hardware and software functionality, and it contains the authentication and key agreement (AKA) functions (symmetric crypto). The SIM also contains the security credentials, like the permanent subscriber identity, the International Mobile Subscriber Identity (IMSI), and the corresponding 128-bit authentication secret, called K_I in the 2G SIM.

The AKA protocol used is called GSM AKA, and it is a single-pass challenge-response protocol with a signed response (SRES). The challenge is a pseudo-random 128-bit RAND bit-field and the response is the 32-bit SRES element. The challenge-response part is dependent on an “authentication set” forwarding stage, in which the HPLMN forwards the authentication credentials to the VPLMN network. The protocol runs between the SIM and the visited network. This scheme is efficient and allows for fast and simple authentication of the subscriber as well as deriving a session key (the 64-bit K_C). The SIM features the A3 and A8 AKA interfaces, which are only found in the SIM and the home subscriber database (HLR). The original example implementation of A3 and A8, called COMP128, is cryptographically broken [13], but still seems to be in use in many markets.

Over-the-air encryption is by means of the A5 stream cipher family, which is located in the mobile phone and the base transceiver station (BTS). There are several A5 versions available, but the original A5/1 is still the default and mandatory-to-deploy algorithm. It can easily be broken today by a dedicated attacker [14]. The breaking of A5/1 is based on a clever variant of applied brute-force and space/time trade-offs called a rainbow table attack. First, one essentially brute-force breaks A5/1 and stores the results in large tables. This is a once-only effort. The process is computationally very costly and also very time consuming, but modern graphics cards makes this feasible and even quite affordable. The process also requires considerable storage (terabytes), but this has become a commodity. Subsequently, one uses the stored tables and clever algorithms to derive the session keys. This second step is fast and computationally inexpensive.

The A5/2 algorithm, which was explicitly designed to be weak (CoCom regulations), is officially deprecated. The A5/3 algorithm, which is based on the 3G KASUMI design, is the current best option for GSM, but rainbow table attacks still work since the algorithm is limited to 64-bit [15]. The A5 family is based around a 64-bit key, expect the recent A5/4 cipher, which is a 128-bit design based on the KASUMI algorithm. In GPRS, one uses the GSM AKA protocol as-is, but here one uses the GPRS Encryption Algorithm (GEA) ciphers to protect the asynchronous packet transfers.

3) Omissions and Shortcomings: There are many obvious omissions and shortcomings to GSM security. This is not strange as the 2G systems do not have a security architecture as such; it is more akin to a collections or measures put together without well-defined requirements. The following list (derived

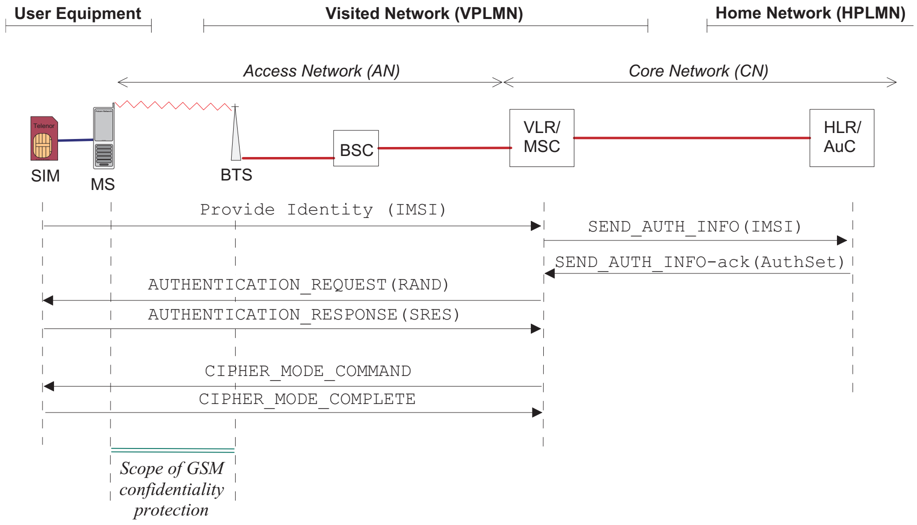

Figure 1: 2G Security: GSM security overview.

in [16]) identifies some of the flaws. Even with all these flaws, the GSM/GPRS system has been a remarkably secure system. However, some 25 years down the line and the shortcomings have become serious liabilities. There are also a number of implementations issues [17]. The list is not fair with regard to the threats found early on, but it is certainly valid now.

- One-way authentication is utterly inadequate.
- Delegated authentication is naive trust-wise.
- SIM/AuC: pre-shared authentication secrets is a liability.
- No inter-operator authentication.
- No way to authenticate system nodes.
- No uniqueness/freshness to challenges.
- Unauthenticated plain-text transfer of security credentials.
- Unprotected key transfer.
- Missing key binding and too short keys.
- Key refresh dependent of re-authentication.
- Missing expiry condition on security context.
- Weak A3/A8 functions and no key-deriving key structure.
- Short A5 key stream cycle and key stream re-use.
- Redundant and structured input to A5 (expand-then-encrypt).
- Highly redundant input to A5 (in signaling message).
- Protection coverage/range too short (only MS – BTS).
- Missing integrity protection.
- Weak/inadequate identity/location privacy.
- No core network control plane (signaling) security features.
- No core network user plane protection.
- No IP layer protection (GPRS).
- No mobile phone (MS) platform security.

B. 3G Security

1) Background and Requirements: Security in the UMTS system is described briefly in [16, 18] and in considerable depth in [19]. The main security specification is TS 33.102 [20]. A “Security Objectives and Principles” [4] background document was also provided, together with a threats and requirements analysis document [3]. One also introduced Network Domain Security (NDS), which includes IPsec profiles for use with 3GPP systems [5] and a standard set of public-key infrastructure (PKI) protocols and methods [21].

2) The 3G Security Architecture: The UMTS security architecture, depicted in Figure 2, is an important overhaul of the GSM security, yet the underlying system model remains much the same. Amongst the features are:

- New subscriber card (UICC) with security module (USIM).
- Introduction of 128-bit crypto primitives.
- Improved two-way'ish AKA algorithm (UMTS AKA).
- Introduction of core network protection (IP protocols).

Sadly, backwards compatibility concerns also dictated that the GSM SIM could still be used, which when used reintroduces many if not most of the 2G weaknesses.

3) The IP Multimedia Subsystem (IMS): IMS came with UMTS (Rel.5). We do not include IMS in our discussions as it is an optional service-level feature.

We note that a cut-down version of IMS will be used to support voice over LTE (VoLTE), and this version (IMS MMTel) will be important in 4G systems.

4) Omissions and Shortcomings: The 3G security is substantially better and more future proof than the 2G security, and one really has a security architecture. The architecture is by no means perfect or complete, but it does at least capture the main risks/threats and defines what one wants to protect. Completeness will always be an issue, but in the 3G systems we also have that there sometimes is a considerable mismatch between stated goal and what the mechanisms achieve. A case

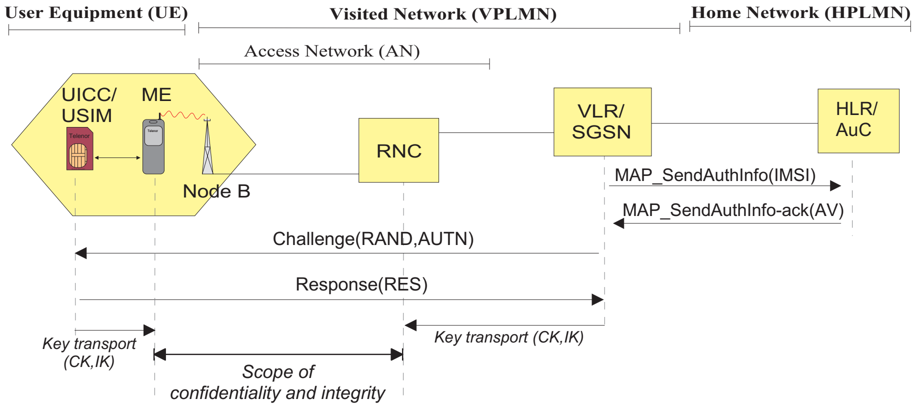

Figure 2: 3G Security: UMTS security architecture.

in point would be the identity/location privacy requirements, which does capture the problem well, but the mechanisms that should provide the necessary services are woefully inadequate. However, they are *a)* exactly the same as for the 2G systems and *b)* intimately tied to the identity presentation scheme defined in the basic mobility management (MM) protocol machinery (discussed in [16, 22]). The identity presentation scheme is weak security-wise on several levels, and there are also effective and efficient denial-of-service attacks against it [23, 24]. These problems cannot be remedied by tinkering and quick fixes as they are inherent to the system access procedures. Making changes to the access procedures would have been a major undertaking, and since there was considerable time pressure to complete the 3G standard, improvements to identity/location privacy simply did not happen (there were efforts investigating the possibilities during the Rel.99 design).

Many of the items on the 2G list of omissions and shortcomings are mitigated and resolved, but suffice to say that many of the 2G weaknesses were inherited or permitted through backwards compatibility requirements. Another main problem with 3G security is the limited scope.

C. 4G Security

1) Background and Requirements: The book “LTE Security” [25] is a good and thorough introduction to the topic. The main security standard for LTE is TS 33.401 [6]. LTE and LTE-A are very similar with respect to the security architecture, which for historical reasons is called the “System Architecture Evolution (SAE)” security architecture. The term Evolved Packet System (EPS) is also used.

The radio access architecture changed significantly with LTE, and this triggered large-scale changes to the whole system, including the security architecture. This, together with wholesale abandonment of non-IP based system protocols, marks a clear cut from previous practices. Despite these changes, the security requirements were retained more or less as-is. For compatibility reasons and due to time constraints during the design phase, the UMTS AKA protocol was retained as a component of the EPS AKA protocol.

A main benefit of retaining the UMTS AKA protocol as a component is that one did not have to introduce a new software module on the UICC. Of course, this is also the main drawback, as this rules out more far reaching improvements to the security architecture. In particular, this ruled out using asymmetric public-key based crypto credentials as the basis for subscriber authentication and it ruled out using a Perfect-Forward Secrecy (PFS) based mechanism for key agreement. In retrospect, both these features are going to be needed and it was an opportunity lost not to introduce them in the 4G security architecture.

2) The 4G Security Architecture: The LTE security architecture has a lot in common with 3G security, but with some important changes. Amongst the LTE features are:

- UICC/USIM is retained and required.
- Introduction of full key-deriving key hierarchy.
- Session keys not dependent on re-authentication.
- Auth. master key (K_{ASME}) bounded to VPLMN id.
- New session keys for every handover.
- Separation of user plane and control plane protection.
- Introduction of improved AKA algorithm (EPS AKA).

A welcome change is that backwards compatibility with GSM SIM is prohibited for access to E-UTRAN. UMTS AKA derived security contexts can be used (mapped) to LTE contexts. Figure 3 depicts the EPS key hierarchy, which is very different from the 2G/3G schemes.

The new key derivations take place exclusively outside the UICC/USIM. This makes for a significant departure from previous practices. It also makes the USIM somewhat less significant, given that the mobile equipment (ME) now takes over that functionality.

3) Omissions and Shortcomings: The list of omissions and shortcoming is shorter for LTE, but there are also new types of threats. In a world of smart phones, it is obvious that 128-bit crypto on the access link may count for nothing if the mobile phone is infested with malicious Apps. Likewise, the networks are often hybrid systems, and it is common to have base stations that are 2G/3G/4G compliant. With different security levels and common hardware/software, it is clear that strong 4G protection may easily be offset with weak 2G/3G protection. For 4G this is quite important, as all eNBs will in principle be able to reach all other eNBs.

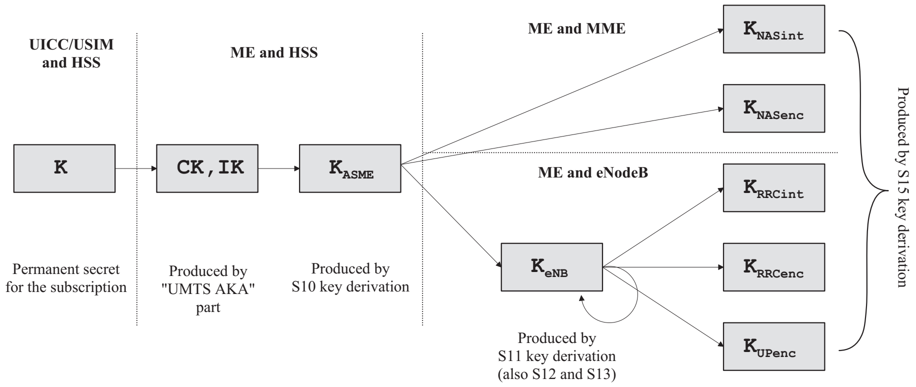
UICC/USIM and HSS
ME and HSS
ME and MME
ME and eNodeB
K
CK,IK
K_ASME
K_NASint
K_NASenc
K_RRCint
K_RRCenc
K_UPenc
K_eNB
Permanent secret for the subscription
Produced by "UMTS AKA" part
Produced by S10 key derivation
Produced by S11 key derivation (also S12 and S13)
Produced by S15 key derivation

Figure 3: 4G Security: The EPS key hierarchy.

Thus, one compromised eNB can reach all other eNBs in the network segment (which may span the entire operator network). It is also clear that many of the nodes, including the base station (BTS/NB/eNB) may be running commodity operating systems (OS). The chosen OS, likely a Linux variant, may be reasonably secure, but even a high-security OS will have weaknesses and must be properly managed to remain secure. Also, introduction of firewalls and intrusion detection systems will be required for base stations. They have become security sensitive servers and must be handled that way.

Server hardening is a must for all network elements, and even so it is clear that not all attacks can be prevented. This means that prevention alone cannot be a viable future strategy.

The EPS security architecture does require the eNB to be secure, but the specification is not very specific [6]. It also has recommendations on use of firewalls, but the specification is quite vague on this subject too. Altogether the systems cannot be said to be fully specified with respect to security. For a greenfield 4G system, the security may be quite good at what the system provides, but the standard system does not do all it needs to do. Also, it is obvious that the user equipment (UE) must be protected. The UE normally is not owned or controlled by the network operator, but it may still be prudent practice for the HPLMN to offer security software to the users. This is not only to protect the user, which a HPLMN should be interested in anyhow, but also to protect the network as a population of broadband devices could disrupt the access network. Distributed Denial-of-Service (DDoS) attacks would be one possibility [26].

D. Some Architectural Oddities and Vulnerabilities

One puzzling aspect of the 3GPP security architectures is that while identity presentation and entity authentication is fully standardized, there is no authorization mechanisms present. There are of course mechanisms to discriminate subscriber based on the type of subscription, but these schemes are not a feature of the security architecture.

Another aspect to be noted is that the subscriber identity that actually is authenticated, the IMSI, is basically a link layer identifier. Since there is only basic connectivity present at the link layer it may help explain why there never was any built-in authorization scheme in the 3GPP security architecture.

As a high-level observation, we also note that the shared-key basis for authentication and key agreement in GSM, and for that matter in 3G and 4G too, is a liability. One is critically dependent on the security of **a)** the production of the SIM cards and one is critically dependent on the security of the HLR/AuC (or HSS) servers. A related issue is the fact that there is no PFS. That is, given knowledge of the permanent secret key (K or Ki) stored at the SIM/USIM and in the HLR/HSS, it is possible to decrypt every session there ever was with that given subscription. This is so because the other key derivation ingredient, the random challenge ($RAND$), is present in plaintext in the session setup signalling, and thus readily available to the intruder.

Thus, if an intruder records all encrypted calls, it can easily decrypt them all later using the secret key. That makes the secret key a very valuable asset and it represents a huge liability to subscriber privacy. The SIM/USIM authentication secret (Ki or K) is a shared secret and it is embedded in the chip at production time. In many cases the personalization of the smart-card is done by the smart-card manufacturer, in which case the secret key will be forwarded to the HPLMN operator. In this scenario, the trust one must have in the SIM card manufacturer is very high indeed. The required trust in the HPLMN is obviously also very high.

If the core key material had been based on asymmetric crypto, one could have let the SIM card generate the private keys themselves. The embedded key material would then never have been exposed, only the corresponding public part would be copied to the HLR/HSS. Furthermore, if the key material would subsequently be limited to authentication only, one could use it to authenticate Diffie-Hellman key exchanges. These key exchanges provide PFS, and is very much a preferred solution in the possible presence of an intruder with the ability to subvert SIM card production. Due to the Snowden leaks, we now know that a major smart-card manufacturer has indeed been hacked [27].

III. Evolving Security Architecture

A. What Kind of Evolution?

It is, of course, obvious that we are not dealing with biological darwinian evolution. The key components of biological evolution, random mutation and natural selection, are not present as-is. In particular, we have absolutely no reason to suggest that “random mutation” is involved in system design. Design decisions may appear arbitrary at times, but they are not random. We have no “natural selection” either, but there is a certain level of selection in the sense that solutions that are too weak or useless, in one respect or another, will be a liability to the system. At the extreme one may think about the Heartbleed vulnerability [28] in the OpenSSL software as a selection case. That particular implementation/version of OpenSSL was in some sense put under strong selective pressure to be removed.

So, the closest one comes biological evolution is probably in terms of diversity of implementation and selection of vendors and operators. Diversity, as a means for protection and changing the attack surface, is being explored as a security measure [29]. Another area where one might be tempted to use biologically inspired comparisons, is the arms race between anti-virus products, firewalls and intrusion detection/prevention products and the attacker tools [30]. Overall, however, there seems to be little evidence that security, which is largely standardized, is an arena for darwinian evolution as such.

With biological evolution we essentially have that every generation *must* be competitive in their habitat. It is not possible to skip generations and add entirely new features. With a designed artificial system it may be expected that one may skip intermediate steps, and move directly on to an entirely new feature or function. However, this is not necessarily the case in practice. The new design is often by step-wise refinement, and requirements for backwards compatibility etc. often makes it prudent not to make radical changes. So, in this way, there is indeed quite a few similarities to natural evolution. However, there will also be many small, and sometimes large, breaks with the past. Still, from the vantage point of the overall system, most changes are evolutionary rather than a clear break from the past.

We note that it is sometimes necessary to be more revolutionary to address problems with deep root causes. Security design occasionally comes into this category.

To summarize, the kind of evolution we are dealing with certainly is not Darwinian as such, but the many of the high-level features may yet appear that way. This is particularly true for very large systems, which will tend to behave in a non-deterministic way and where there is no absolute authority to control the system. There may be design authorities and there may be authorities on other aspects, but ultimately these only control a part of the overall system. For instance, with the internet one have multiple authorities control various aspects of the design, the protocols, the address allocation, etc., but nevertheless, none of these authorities have jurisdiction and influence over the overall internet usage as such.

B. Time-line and Security Goals

One immediately obvious observation is that along the system time-line the security goals will evolve with the target system. To maintain isoquant security, the security architecture must provide services that match the developments in threat level and asset values. This means that while the security goals may have long-term validity, the security architecture services must evolve with the greater system context and importantly be able to address new threats. To not improve on the existing security will almost certainly mean that the security level drops. This decay in the system security level is somewhat reminiscent of increasing entropy, and it highlights the need to spend energy/effort just to maintain the current security level.

This also means that if one wants to expand on the scope or provide actual improvements, one “must run at least twice as fast”. In Lewis Carrol’s “Through the Looking Glass” the Red Queen summarizes this quite nicely.

> “Now, here, you see, it takes all the running you can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that!”
>
> – *The Red Queen [31].*

C. The Triad of Protection, Detection and Response

With the 3GPP systems, the protection designed into the security architecture is by and large of a proactive nature. That is, the schemes are focused on the classical security features of entity authentication, data integrity and data confidentiality. These are clearly needed features, and they are provided to ensure that only authenticated and authorized entities will have access to system resources. Additionally, there are concessions to subscriber privacy. This is all well and good, provided that the schemes are sufficiently strong, and that they are consistent and complete.

However, in a large-scale system system, security breaches and incidents will happen with statistical certainty. This is certainly true for the 3GPP systems, which suffers from weak schemes (for GSM in particular), inconsistent security levels and incomplete protection across the system. Obviously then, there is also a need for handling security breaches.

So, we can safely assume that a security architecture must include both proactive and reactive security measures. The proactive (*protection*) measures would be the baseline protection schemes and would include entity authentication, authorization schemes, data confidentiality, data integrity, etc. Traditional server harding would also fall into this category, including the all too familiar administration of security patches and updates.

However, for large system, it is inevitable that there will be security breaches. We can thus postulate that there will be security incidents. With this in mind, it is prudent to be able to handle this. Firstly, one must of course be able to detect that there has been an incident. Obviously then, intrusion- and incident *detection* must be regarded a security architecture requirement.

Of course, detection alone is not enough. One must also handle the detected events, and reactive security measures must be available in the security repertoire. These *response* measures must be fairly flexible since we generally cannot predict what kind of incidents one must be able to handle. To this end, it is both useful and probably necessary to keep human operators in the response arsenal. Humans, while capable of being flexible, are obviously quite slow in the context of an automated attack. Therefore, it seems prudent to have automated response schemes too.

D. Qualitative and Quantitative Aspects

Some security mechanisms are designed to be *secure* as-is. In that sense, they are like a mathematical expression; either true or false, and usually no middle ground. Now, even if we assume that the base security primitive is indeed fully secure, it is all to often the case that the actual security is compromised by wrongful usage, broken assumptions, erroneous implementation, etc [32].

So, essentially, this means that one cannot rely on a single protection scheme. This is true irrespective of the apparent strength of the primary protection. Thus, there is a need for defense-in-depth. The additional protection schemes may also serve to be backup schemes.

With this in mind, it is likely to be cost-effective to provide defense-in-depth coverage for the assets. That is, provided that these auxiliary schemes are part of an overall design and not ad-hoc schemes bolted onto the architecture. Ad-hoc designs, while they may provide a momentary benefit, is likely to incur future overhead in security maintenance and management. They may also prevent better and more appropriate schemes from being developed and deployed, since they may appear to be effective and efficient (while possibly being neither). Still, we advocate a quantitative defense-in-depth approach to security architecture design.

E. Completeness and Resilience

Defense-in-depth schemes may also serve to add security coverage to areas where the primary schemes may not provide adequate protection. This will reduce the overall vulnerability exposure. Added coverage and multiple layers of security will also provide an opportunity to increase the attack resilience, but we must not be naive here. Only with well-designed defense-in-depth strategies can we hope to achieve an actual improvement. Also, it must be noted that when facing dedicated intruders, simple minded auxiliary security schemes may count for nothing. That is, we must differentiate between protection against advanced persistent threats and protection against unsophisticated attacks by opportunistic intruders [33].

Completeness, whether by means of auxiliary mechanisms or not, is clearly an important goal for a security architecture. This has the implication that for any new system feature or new system assets, one must carefully investigate whether of not the existing security will fully cover it.

Resilience and robustness is likewise very important. For instance, there should be no easy way to disable security schemes by provoking the system into fallback-mode. Fallback modes are all to often a business requirement, but one must take all precautions possible to ensure that an attacker cannot abuse such schemes, or at least to avoid and mitigate serious incidents.

F. Why Low Efficiency May Be a Good Thing

If all security schemes are highly optimized we run the risk of losing flexibility. That is, "high efficiency" protection may be excellent against well-known run-of-the-mill attacks, but they may fail against new and novel attacks.

Generic and flexible protection schemes, may appear a bit "extraneous" and be in some way be less efficient, but they can actually provide protection against new and novel attack. We do not claim that such schemes are necessarily more effective than other schemes, but it is useful to keep in mind the difference between effective and efficient. Therefore, we shall advocate a certain level of security redundancy and diversity, but we note that this must be based in design and that the redundancy and diversity must not be shallow in this respect.

This also means that we must not fall for temptations to deploy lightweight security as the primary protection if there is any reason at all to think that this protection will not be sufficient, consistent or complete in the longer run. As an example, if two-way authentication will be needed in the foreseeable future, then deploying a one-way scheme like the GSM AKA protocol will only hurt the system architecture in the long run. A true two-way authentication protocol may be slightly more expensive to run, but it is future proof and it avoids complications with the need to modify and update it. Likewise, the choice not to use asymmetric cryptographic primitives as the basis for the 3GPP AKA protocol functions may be defended on the grounds that symmetric methods are, computationally, much cheaper to run. However, the symmetric methods are unsuitable for providing PFS, and we know now that lack of PFS is a practical liability [27]. With hindsight, to chose the "low efficiency" asymmetric cryptographic primitives would have been a much better solution than the apparently more efficient symmetric key alternatives.

G. Planned Deprecation

A lot of protection schemes are used even though they are not very secure anymore. Defense-in-depth is one thing, but keeping protection schemes that no longer provide protection is wrong. So, when a crypto primitive is no longer secure one should plan how to deprecate and replace it.

With the 3GPP example systems, it is easy to see that for instance the 64-bit A5 algorithms are no longer future proof. In fact, they are all too weak already. To replace the A5/1 cipher will not be easy, and it is a process that will take considerable time, given that the A5/1 algorithm plays a crucial role in inter-operability for roaming subscribers. However, this only highlight the need to plan ahead and to start the process. When the A5/2 algorithm was deprecated it literally took years before it was officially an algorithm non-grata. And this was for an auxiliary algorithm.

Planned deprecation also implies planned migration, and there may be cut-off dates, etc., involved. This will never be easy to accomplish, but unless one initiates the process early on one should brace for the impact when change is forced upon the system. With this in mind, one should always include a "best before" date on all security components and mechanisms. This was actually done for the KASUMI cipher [34]. The actual statement made in the evaluation report was that the algorithm should be reviewed every five years to verify the security and usability [35]. In practice, this meant that KASUMI was deemed safe enough for 3G security but unsuitable for 4G security. It also meant that 3GPP commissioned the development of an alternative to KASUMI, the SNOW-3G algorithm [36].

The lesson is that one should plan for the schemes eventual deprecation during the design process when the scheme is included in the security architecture.

H. Facilitating Secure Migration

As stated above, deprecation of a security scheme will tend to imply migration to a new scheme. When this is the case, it is of course imperative that the migration process is secure. This can be quite difficult to achieve, since it is more than likely that the old and new scheme must co-exits for a considerable time. This again implies a capability to securely negotiate the right scheme. It also implies a well-though out migration plan and security policies that matches this.

As a fact of life, one may also need fall-back from a newer security scheme to an older scheme. This usually implies going to a lower security level. The triggering conditions may include incompatibilities between the negotiating entities, but whatever the fall-back decision is based on, one must make sure that an intruder cannot trigger this condition too easily. We casually observe that a legitimate system entity may also be an intruder. Fall-back options are messy and very hard to make secure. To allow them means accepting higher risks, and fall-back solutions should certainly be monitored and they should certainly have their best-before dates. It is of course also essential that they are captured in the security policies.

The requirement for secure negotiation of security schemes must necessarily mean that there is a security basis to facilitate the negotiation. This security basis must be valid in a long-term perspective and it must be rock solid and fully trusted. Efficiency is not a primary requirement here, and it is akin to a root certificate in a PKI system. The root certificate may have excessive key length and use computationally inefficient algorithms, but this does not matter since it is used infrequently and since rock solid security is the only real imperative.

I. Mitigation and Recovery

Mitigation and recovery is in many ways part of the *response* requirement. However, we want to make it explicit that schemes that exclusively facilitate mitigation may have a place in the system. These schemes merely reduce the impact of an incident, but that may be a worthwhile goal and it may also be a cost-effective option.

Recovery schemes will obviously also be needed. These are after-the-fact schemes that simply aims at restoring operation after an incident. Needless to say, initiating a recovery operation must be subject to authorization.

J. The Scalability War

The classical Dolev-Yao intruder model is not the most realistic intruder model [37]. Real intruder will use any available means (subversion, physical intrusion, tricking the principals), ultimately being as powerful as a Dolev-Yao intruder. An National Security Agency (NSA) type of intruder will obviously also use the legal procedures to get access to systems. There is a reasonably body of papers detailing various intruder models, but suffice to say that a modern CI system must be able to handle **all** types of intruders. Furthermore, the CI system, inevitably exposed by having an internet presence, must face the prospect of distributed attacks. Distributed Denial-of-Service (DDoS) attacks are not new, and they may also be initiated over wireless connections [26, 38]. Other types of distributed attacks are also possible, and they may actually use DDoS attacks as a means to trigger error conditions, which then are exploited.

This inherently means that the system *must* have efficient as well as effective protection, and that mechanisms that do not scale well, compared to intruder capabilities, will be doomed to fail in the long run.

Our assumptions related to scalability and efficiency:

1) Security scalability will be a major concern.
2) Efficiency is highly important.
3) Effectiveness is imperative for core mechanism.
4) Auxiliary defense-in-depth solution are needed.
5) Avoid specific-attack measures if at all possible.
6) Security management must scale well.

See [39] for some considerations concerning scalability in general in the world-wide web context.

Assumptions three and four are apparently somewhat at odds, but in the end assumption three can be supported given that these means are complementary and cost-effective. See also considerations about the economy of attacks and defenses outlined in [33], This indicates that for broad sweeping attacks, even quite weak mechanisms may successfully thwart the attacks. Measures that are only effective for one specific attack should generally be avoided.

One must be able to handle a multitude of opportunistic, but probably not too capable, intruder and one must provide reasonable protection against capable intruders. There is also a significant difference in those attacks that scale effortlessly and those that do not. Defense schemes whose sole purpose is to increase the attack cost may therefore have a justification.

IV. Why Change the Security Architecture?

The short answer is that we need to change the security architecture because some of the premises for the original security architecture have changed. A slightly longer answer would revolve around the following aspects.

A. High-level change triggers

There are many high-level change triggers, amongst others:

- *Changes to the assets of the system.*
  This could include changes to the value of the existing assets, inclusion of new assets or removal of assets.
- *Changes in the threats towards the assets.*
  This includes assets exposure, new intruders, new intruder capabilities. For new assets it could also include missing or mismatched protection.
- *Changes to the system context.*
  The system may initially have played a limited role, but may have evolved into something more.

The engineering aspects of security design and implementation are not new [40], but likewise it is not exactly new either that there may often be a mismatch between the design requirements and the real-world threats and needs [32]. For the 3GPP systems, it is quite clear that the financial value of a network operation has increased sharply during the lifetime of the 3GPP systems. That is, there are many orders more of subscribers than there originally were. The assets have similarly evolved such, and not surprisingly, the threats towards the systems have changed substantially over the years.

B. Evolution aspects

Large-scale long-lived systems cannot remain as static objects for long. Instead, they must be dynamic and adapt to changing environments. This is true of the 3GPP systems too. A network operator that only provide speech and short messages will not be as attractive as operators with a more complete set of services. Price will to some extent influence this, but then one may see a lower relative price as a change to the value of the assets, and as such it is in some sense an adjustment to a changing environment.

- *Evolving Target System.*
  If the target system changes, then this will likely affect the security architecture. Still, the nature of the change may be such that it does not trigger a need for updating the security architecture.
- *Evolving Security Architecture.*
  The security architecture may need updates and modifications due to external circumstances, or even completion of planned features that were not initially fully specified. Changes in the threats towards the assets, the exposure of the assets, and the number of users will also affect the system. It could also involve changing trust-relationships and changes to value of the assets. All these are at play with the 3GPP systems.
- *Security Evolution History.*
  An evolving system is obviously a product of its history. Decisions taken during the design of GSM still have an impact on LTE. For instance, the basic identity presentation scheme essentially remains the same for LTE as for GSM [41, 42].
- *Societal Impact.*
  When a system reaches certain thresholds it will take on a new role. It enters a state of criticality to society and will become an object of regulatory interest. The critical infrastructure (CI) requirements will focus on system survival and service availability rather than security and privacy for the individual.
- *Privacy.*
  Privacy requirements may not have mattered too much for a small system with few users back in the early 1990ties. Today, privacy requirements are often mandated by laws and regulations [43].

V. Assumptions Regarding Systems, Security and Cryptographic Characteristics

The following set of assumptions not all be true for all systems, but we advocate assuming that they are true.

Some of the assumptions are relatively self-evident in nature, while others may appear less justified. Nevertheless, the value of these assumptions is more as guidelines to a design process than as propositions that must be defended.

A. Assumptions about Successful Systems

We assume that when people start to design a system they intend it to be successful. Thus, they must take the above into account in their design. Our high-level assumptions about a successful system:

1) It will outlive its intended lifetime (and design).
2) It will have many more users then originally intended.
3) It will need to scale its services cost-effectively.
4) It will become highly valuable (many/valuable assets).
5) It will outlive its base technologies.
6) It may become a critical system (company, organization).
7) It may become a critical infrastructure (society-at-large).
8) It will spawn unsuccessful branches/features.
9) It will have to deal with multi-vendor cases.
10) It will need to operate with multiple releases in place.
11) It must encompass all of operations & maintenance too.
12) It will be subject to regulatory interventions.

B. Assumptions about System Security

Our assumptions about a long-lived security architecture:

1) The assets will change (value/number/types).
2) The principal parties will change and multiply.
3) The threats will change.
4) Trust models will fail (and/or become outdated).
5) Trust will be betrayed.
6) Risk evaluations will be outdated.
7) Weaknesses, vulnerabilities and exposure will change.
8) Intruders will become more powerful and proliferate.
9) Attacks will only be better over time.
10) There will be security incidents.
11) Scalability in security mechanisms will be decisive.
12) No single security scheme or approach will suffice.
13) Effective and efficient defense-in-depth will be needed.
14) Pro-active security protection will not be sufficient.
15) Re-active security will be very important.
16) Ability to handle large incidents will be required.
17) Deprecation of security schemes must be built-in.
18) Secure fall-back must be supported (but not trusted).
19) Security negotiation must be built-in.
20) Mitigation and recovery must be supported.
21) Pervasive resilience and robustness is required.
22) Autonomous response will become important.
23) There will be security architecture omissions.
24) There will be security issues (multi-vendor).
25) There will be security issues (multi-release).
26) Fixing minor security wholes can take a very long time.
27) Fixing the security architecture may take years.
28) Security management will be crucial.
29) Security configuration management is crucial.
30) Security migration methods should be built-in.
31) Security policies will be inadequate and incomplete.
32) Security policies will be outdated.
33) Privacy will become ever more important.

This list of assumptions should not be read as a definitive or authoritative list, but rather as a starting point.

C. Assumptions about Cryptographic Solutions

Our assumptions related to cryptographic solutions:

1) The cryptographic base functions must be future-proof.
2) Cryptographic primitives will be broken (or become too weak).
3) Key sizes will be changed.

4) Security protocols will be broken (or become too weak).
5) Cryptographic parameters will need to be negotiated (securely).
6) Cryptographic primitives will need to be revoked.
7) Implementations will contain weaknesses.
8) Management of cryptographic elements will be crucial.

It is clear that the basic boot-strapping fundament must be very solid. This minimal base is what you will depend on if you need to boot-strap new security solution and new cryptographic primitives in the rest of the security architecture. It needs to contain enough to support boot-strapping and it needs to be future-proof. Efficiency is *not* a main priority here.

VI. Too Big to Fail?

Even very large systems can, and almost certainly will, fail at some point in time. Consider the collapse of the Soviet Union [44], the bankruptcy of the Lehman Brothers Holdings bank [45] or for that matter the rise and fall of the AltaVista search engine [46]. In all three cases, these were large and powerful entities in in their respective domains.

System failure may be temporal, partial or spatially confined. It may also be permanent, complete and global. In this section, we take a look at some of the factors that may contribute to failure. Much of this section will be conjecture. The purpose is not to derive grand conclusions, but rather attempts at understanding a little more about some of the factors that come into play. That, and pointing to areas were we need more research.

A. Evolution and Architectural Decay

Big systems are complex entities, and security architectures no less so. There is, at least initially, a high degree of structure in how the architecture is organized. The complexity one finds will therefore tend be be necessary complexity.

Evolution implies changes, and unless meticulously executed, the changes will complicate the architecture. Some of the complexity will then tend to be a product of the change process itself. The complexity will increase, but the structure may actually be less clear. In short, there will be increased entropy. In thermodynamics, the entropy increases due to random changes to a system with a high degree of structure. The changes will have (with statistical certainty) a higher likelihood of distorting the structure than improving it.

Designed evolution is not of course random by nature. Still, with many different and competing requirements, it is to be expected that some of the design decisions will be sub-optimal or counter-productive with respect to maintaining structural consistency. In a highly organized system architecture, this will inevitably lead to a less structured, less coherent and less consistent system over time.

Requirements for backwards compatibility will complicate matters, and this is almost always something that will lead to less structure and/or less consistency. Given that a designed evolution is not random, one may expect that many of the design decision will actually improve the structure. Thus, there is a counter action to the architectural decay.

For the 3GPP systems, it is essential that backwards compatibility with older and insecure security schemes is deprecated to avoid architectural decay. To some extent this is happening, and the fact that the GSM AKA protocol cannot be used for authentication and key agreement in LTE is a sign of that. However, it is essential that the rate of deprecations and obsolesce does not lag too far behind the rate innovation and new schemes.

B. Black Swans

The theory of black swan events describes black swans as something completely unexpected, yet with hindsight it appears much more predictable. The concept is derived from the fact that prior to discovering Australia, Europeans simply could not envisage something like a black swan.

The concept has been popularized by Nassim N. Taleb [47, 48] and are associated with financial events. The Lehman Brothers collapse was a black swan event in this sense. Taleb defines black swan events this way:

1) The event is a total surprise.
2) The event has large and even severe effect.
3) The event was later, with hindsight, seen as predictable.

In terms of the financial systems background, Taleb attributes 1) to a failure of understanding the statistics properly. This in part is due to not understanding the nature of randomness and not understanding that statistical distributions simply are not well defined for singular or very infrequent events. That is, you cannot reliably determine the confidence interval, deviation or frequency of a class of events with little or no historical data. This, in effect, means that you cannot rely on statistics to predict those events since you do not even know the distribution. As for point no. 3, it usually seems clear in retrospect that the risk was always there. Given new knowledge, it will even seem obvious that the black swan event occurred. This is of course the benefit of hindsight, and can to a large degree be attributed to what Taleb describes as our inability to acknowledge the role of randomness. When the result is known then it is indeed no longer a surprise.

In terms of security and security architectures a black swan event would be something that simply is not captured at all. This could be due to the magnitude of the event(s) or down to the combination of events.

It could also be down to events that simply are not captured by the system model or down to our lack of understanding what the real system threats are. This is in particular a risk for evolved systems. In our case study object, the 3GPP systems have a number of both standardized and non-standardized security measures. These have evolved over the years, and have grown to address most of the perceived and experienced threats. But, at the same time the exposure of vulnerabilities/weaknesses change and the attack surface probably increases, with or without being acknowledged. Most of the security measures are inadequate in the sense that they do not stop or fully address the threats, but they do impede and/or mitigate the threats such that the risk is low (or believed to be low).

Part of the security risk problem is that we generally do not understand threats or risks very well. People, even professionals, are not good at foreseing which threats are realistic and not, we fail to foresee impact and we do not really understand scalability. The last part is important, since scalability is what ultimately may break a system. A successful attack on 10,000 cellular subscribers is bad in many ways, but if an attack is limited to that level it will not pose a threat to the cellular networks as such.

In the same sense that we do not fully understand how attacks scale, we do not really know how defense mechanisms scale either. Even a weak defense-in-depth auxiliary scheme may be effective on a system level. It may not be effective against advanced persistent threats, but could fend off broad-scale automated attacks. Even system diversity options that were not intended to be security mechanisms may contribute here since they alter the attack surface [29, 33].

C. Why Don't We Listen to Warnings

In Greek mythology, Cassandra was a daughter of Priam, the King of Troy. Cassandra was a very beautiful lady, and she attracted the attention of Apollo. He provided her with the gift of prophecy, but got vengeful when Cassandra refused his romantic advances. He then cursed her so that nobody would believe her warnings. So, while she could foretell future events, she had no way of altering the events or convince others about future perils. This is why her warnings about the Trojan horse went unheeded.

Warnings about inadequate security or of new and potentially devastating threats come and go. There are so many security inadequacies and yet they do not seem to cause severe problems. This apparent paradox is explored in the paper "Where do all the attacks go?" [33], and part of the answer seems to be that most attacks are unsophisticated and opportunistic by nature. The intruder go for the low hanging fruit and do not necessarily target a specific system or host.

In the 3GPP realm, we have examples of appallingly weak security and yet the protection somehow seems adequate for the purpose. The GSM authentication is one-way only, the encryption is only 64-bit wide, there is no integrity protection, and yet GSM seems adequately safe for what it is. Still, there is of course a tipping point there somewhere, where attacks get practical (this has happened) and where the cost of doing so gets sufficiently low to allow everyone to do it.

There are many people warning about the GSM weaknesses [13–15, 23, 27, 49], and one might even describe this as a chorus of Cassandras [50]. However, if a catastrophe does not occur on schedule, we tend to discount the messenger. Cry wolf to many times and people get tired of the message and the messenger.

The complexity of a modern critical ICT system is daunting, and we cannot blame even the system architects for not fully understanding it. Security is in many way even harder to understand as it is fundamentally about missing, incomplete or inadequate functionality in a given context (and where the context is continually subject to change). So, top level management must learn to live with false alarms (noise) and will have to dismiss them regularly. This, of course, makes it only harder for an important warning (signal) to get through.

That is, security architects must learn to live with warnings not being heeded.

D. Unprepared and Unaware of It

In the paper "Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments" [51], the authors investigates the ability of people to assess their own competence. While this may be culturally dependent and while it may not apply under all circumstances, the paper highlight the fact that incompetent persons do not necessarily recognize their incompetence. Since they know so little they have no way of knowing what they do not know.

Curiously enough, the opposite seems also to be true. The best skilled people often do not recognize how good they are, since they tend to compare themselves to other highly qualified people. The end result is that they see themselves as more average than they really are.

For large security architectures like the 3GPP system, we have that there is no single person or individual company that has full design authority of the system. We have, as noted earlier, the 3GPP system represents a clear case of contribution-driven design-by-committee regime. The individuals or these committees may or may not be unskilled or unprepared, but one may speculate sometimes if the organization of the specification work is such that the design appear to have been done by an unskilled architect. There may certainly be awareness of this on the level of the participating individuals, but it is not at all clear that there is awareness on the organizational level.

In terms of security, we have that it is generally very difficult to correctly assess threats and risks. It is also very difficult to assess what good security really is. The Heartbleed incident comes to mind again [28]. This was not a design flaw as such, but implementation errors do happen and one must be prepared to handle incidents whatever the cause. Too many organizations were unprepared for something like this, and moreover, they seemed unaware of their unpreparedness.

This will easily lead to situations where one invest unproportionate time and money on features that may in the end only prove a bare minimum of protection. Those schemes may be important as-is, but one the whole the balance is uneven when considering where time and money was spent. The problem is even more acute for security architectures than for single schemes. So, it is conceivable that we face a situation where the very best experts do not assert themselves as they should have and where people without real expertise may exert considerable influence.

The unskilled/unprepared paradigm may also accentuate the so-called "Bikeshedding problem".

E. Painting Bikesheds

Parkinson's "law of triviality" is commonly referred to as bikeshedding. The law is associated with Parkinson's 1957 observation that organizations give disproportionate weight to trivial issues [52]. The "law" emerges from a case-study of a committee whose job it was to approve plans for a nuclear power plant. The committee spent a lot of its time on trivial and unimportant issues. These issues were easy-to-grasp and did not require insight, preparation or deep understanding.

Amongst the items discussed was material choice for the material to be use for the staff bike-shed. At the same time, the committee almost neglected discussing the proposed design of the nuclear power plant itself. Of course, that design was complex and it would have taken a real effort to comment on it with insight,

The term “bikeshedding” was further popularized with the email posting of “A bike shed (any colour will do) on greener grass...” to the FreeBSD mailing list [53]. The author, Poul-Henning Kamp, cites the discussions about the updating of a minor function in FreeBSD, and then goes on to explain that:

> A bike shed on the other hand. Anyone can build one of those over a weekend, and still have time to watch the game on TV. So no matter how well prepared, no matter how reasonable you are with your proposal, somebody will seize the chance to show that he is doing his job, that he is paying attention, that he is *here*.

The bikeshedding problem has of coursed been recognized and there have been attempts to mitigate the effect. Many large-scale system design efforts have an individual as the ultimate arbiter for cases like this. These individual have authority to make final decisions and to stop useless discussions about unimportant features or minor aspects. For instance, in the Linux world, the original designer, Linus Torvalds, have more or less absolute authority over what code is included in the Linux kernel. For the programming language Python we have a similar situation, where the original creator, Guido van Rossum, is appointed Benevolent Dictator For Life (BDFL). We note that an organization like the 3GPP does not really have a any person with ultimate design authority. The closest one comes is the plenary high-level design forums, but these do no give directions or guidance needed to avoid bikeshedding as such.

Given that security and security architectures are indeed very complex entities, we should not be surprised to see a fair amount of bikeshedding here too. This will add to the noise and divert attention of decision makers, so it is important that we are aware of this effect.

F. Inverse Bikeshedding

Complementary to the “law of triviality”, we have another phenomenon that sometimes surface, namely what we term the “inverse bikeshedding” phenomenon. This phenomenon is more or less the opposite of “bikesheeding”, and here we have an obsession with attention to highly complex technical details at the cost of ignoring the larger picture.

This is the kind of trap that very clever specialists may fall into. For instance, in the Crypto Forum Research Group (CRFG) associated with the Internet Engineering Task Force (IETF), one could witness heated debate over the elliptic curves used in cryptographic primitives and signature algorithms in spring 2015 [54]. Now, within the context of CFRG it makes sense to focus on cryptographic detail, but it seems that many of the participants fail to see that the actual adoption of an elliptic curve is unlikely to have any relation to the minor differences between the discussed alternatives. Local to the group, the discussion is valid and on topic, within the overall IETF context, the discussions are acceptable if they quickly lead to actionable decisions. Should the discussions not lead to timely and relevant conclusions, then they would appear to be “inverse bikeshedding” activates instead.

It is a leadership challenge to avoid “bikeshedding” and ”inverse bikeshedding” activates, and the role of the group chairs is instrumental to reach timely conclusions.

G. Security Theater

After major incidents there is a need to be seen to “do something”. Thus, not only is there a strong incentive to point out who the bad guys are, but also to come up with measures that appear to counteract the newly discovered (or newly acknowledged) threats. The 2014 hacking of Sony [55] emphasizes the hunt for a culprit. In other cases, we see disproportionate and even completely misguides responses, and we saw quite a lot of that in the wake of the September 11 2001 attacks on the Twin Towers. Bruce Schneier is generally credited as having coined the term security theater and in the book “Beyond Fear” [56] he elucidates the concept. The concept is further refined in [57].

Security theater is not all bad. In particularly, it may offset over-reactions to incidents and allow business to continue as usual where fear would otherwise dominate too much. One may view this part of the 9/11 response as a measure against fear. Since to instill fear is a major goal of terrorists, the illusion of security theater can be seen as a counter-measure to illogical fear of terrorism.

However, security theater is also wrong. Part of this problem is a that one tends to end up with a lot of attention to strengthen unimportant features, often combined with a strong and narrow focus on details. This will not improve actual security very much, and it will in many ways be counterproductive as it diverts resources and attention onto trivial matters. As such it will foster more bikeshedding and a false sense of security.

H. False Security and Cargo Cult Security

Security theater may over time develop into the more elaborate *cargo cult security* type of deception. Then the main functions and mechanisms may all be there (or mimicked closely), but with some vital part missing or done completely wrong. Cargo cultism is defined by “perfect form”, but it simply does not work as intended. Feynman has an amusing description of “cargo cult science” that nicely illustrates the principles [58]. Since security can be very difficult to get right and to verify, cargo cult security may look like the real deal.

Within the 3GPP security architecture one would be hard pressed to find cargo cult security, but if one looks at the wider picture with deployed networks one may find both false security and even cargo cult security.

It is worth noting that those that champion cargo cult security may not recognize that they do so. Either way, cargo cult security is antithetical to real security and may lead to a false sense of security. To do something right is not enough, one must also do the right thing.

I. Trust and The Tragedy of the Commons

The article “The Tragedy of the Commons” [59] is often cited and is an example of a game theoretic problem in which individuals acting independently and rationally according to each’s self-interest, behave contrary to the best interests of the whole group by depleting some common resource. The commons in questions was originally about unregulated grazing rights on common land, but it has application for any common resource accessible to many parties. The problem is an optimization problem, in which the best long-term strategy would be for the individuals to behave cooperatively. However, if enough individuals defect, then it no longer pays out to stay loyal and the best strategy would be to defect.

Security and security architectures are not really a commons resource, and the tragedy of the commons does not necessarily play a direct part here. However, it can be seen to play a part if anti-social attitudes and downright theft becomes the norm for a large enough subset of the population of the users. This will seriously affect the trust climate and systems and societies needs trust to thrive [60]. Without trust, many if not most, transactions would be much more cumbersome and much less effective. There is a soft side to trust and there if a hard side to trust. The hard side consist of methods for enforcing trust and requiring trustworthiness. Security procedures will be amongst the more important ones in the arsenal of hard trust.

So, if soft trust is not seen to pay off, one will often react with tougher hard trust requirements. This could be well justified, but increasing the security level will often have consequences for usability and the transaction costs. This will ultimately have an impact on how efficient the system is.

J. The Somebody Else's Problem and Bystander Apathy

Somewhat related to the tragedy of the commons problem, we have the so-called "Somebody Else's Problem (SEP)". The SEP is humorously explained in the novel "Life, the Universe and Everything" in the "The Hitchhiker's Guide to the Galaxy" books by Douglas Adams [61].

> An SEP is something we can't see, or don't see, or our brain doesn't let us see, because we think that it's somebody else's problem....
> The brain just edits it out, it's like a blind spot. If you look at it directly you won't see it unless you know precisely what it is. Your only hope is to catch it by surprise out of the corner of your eye.

The SEP is understood as a phycological perception problem, in which nobody feels responsible for addressing a particular problem because it is seen as somebody else's problem. The SEP effect is not of course limited to security and security architectures, but it does certainly have an effect here too and it may effectively prevent obvious weaknesses and threats from being addressed.

The somebody else's problem is related to the so-called *unresponsive bystander problem*, named after the case of the killing of Kitty Genovese, in which there were no less than 38 witnesses to the stabbing [62]. As it turns out, people become less responsive to a problem if they do not perceive it as their problem. The feeling of ownership of the problem is substantially weakened when there are other persons present. So much so, that the feeling of responsibility seems to vanish more or less completely when more than four persons are present [63].

Security problems may sometimes come in the SEP/Bystander category. The problem is exacerbated by the fact that sometimes one cannot easily place the responsibility for a problem. Is the given issue a design problem, is it an implementation problem, or is a deployment and configuration problem? This problem is localized in the sense that the SEP apathy may affect anyone, and holistic in its negative effect on the overall security architecture.

Clarifying responsibilities will certainly help with the SEP/Bystander problem, as it reduces the number possibly responsible "bystanders". Organizations like the IETF and the 3GPP have at least some shallow mechanisms in place to that may address the SEP/Bystander problem. There are styleguides on standards documents dictating separate security sections and there are requirements to check for security impacts on new functionality, etc. Still, the SEP/Bystander problem requires organizational awareness to correct it, and it is a leadership responsibility to see that the problem is addressed. We note that there are elected officers and chairpersons in for instance the 3GPP and the IETF.

K. Inefficient Enforcements and Susceptible Parts

The "The Byzantine Generals Problem" [64, 65] is a well-known problem generally concerned with fault tolerance. The Byzantine problem has its background from the Byzantine military, in which each division is controlled by a general. The generals communicate by messengers. Some of the generals will be traitors. How then, in the presence of traitors, can the loyal generals adopt a good plan? The question can be loosely translated into "How many components can we tolerate to fail".

Component failure in large-scale system is not a matter of if, but when and how often. Accidental component failure is one thing, but component failure due to attacks and subversion is another and more serious problem. In a large system, no matter how good the security architecture may be, we will experience weak links. Some of those will be substantially weaker than what the security architecture mandates. That is, we have *inefficient enforcement* of security requirements. Or, of course, the requirements itself may be missing. There are a huge number of reasons for this being so, but rest assure that this condition will affect a certain number of the population of nodes, components and parts in the overall system.

Single failures are probably not problematic at the system level, but we will likely have Byzantine conditions in the system. The security version of the Byzantine problem goes beyond the fault tolerance version in that the problem is more severe. The common part is that below a certain threshold the system cannot be made reliable or secure anymore.

The only viable way to handle this problem is to have good and well rounded detection and response mechanisms in place, together with various redundancy schemes. Good redundancy schemes will improve the system resilience and robustness, but one must ensure that the redundancy is effective and this requires verifying the appropriateness of the redundancy schemes regularly. For instance, if flooding is the problem then placing a redundant server in the same location as the main server is unlikely to be a good solution. In the 3GPP systems, redundancy is not normally part of the design. At best, one has catered for the possibility. This means that redundancy must be part of the implementation, and part of the operation and maintenance of the deployed system.

L. Thresholds and Tipping Points

The Byzantine Generals Problem does point to the fact that there are thresholds, beyond which the system breaks down. Too many traitors amongst the generals, and the decisions process can be effectively subverted. For a large systems, local breakdowns or temporally confined outages are something which one routinely will have to handle. These will not break the system, but they will add to the burden and complexity of maintaining the system.

However, there will inevitably also be breakdowns that cannot be handled so easily. There will be global thresholds and much like in chaos theory, the system may actually look reasonable stable while it is located within its basin of stability [66]. In chaos theory, nonlinearity effectively prevents long-term predictions, even though the system may be mathematically deterministic in nature. That essentially implies, if the system, for one reason or another, strays outside the basin of stability, the outcome will largely be completely unpredictable. This being said, there are also progress towards anticipating these critical transitions [67]. With or without anticipation, there may be system-wide tipping points, after which there are no obvious recovery anymore. That is, recent research points out that recovery, at least in living systems, seems to be related to the "distance" from the tipping point [68]. This provides a glimmer of hope.

Finally, we note that the system may be seem very stable and appear to be highly resilient while within its basin of stability (*basin of attraction*).

These insights are not easily absorbed in security architecture designs, but one lesson seems to be that while one may be unable to prevent such incidents, one may be able to respond to them. This type of "Black Swan" incident would be very hard to predict and the response would be equally hard prescribe a priori. This can be seen as an argument for redundancy and deep diversity in terms of mechanisms available in the response repertoire. It may also be construed as an argument in favour of keeping skilled humans in the loop. It is an argument for emergency response rehearsals and preparedness in general.

M. Unknown Unknowns

The phrase "unkown unknowns" comes from Donald Rumsfeld's perhaps most famous statement while serving as George W. Bush's secretary of defense:

> As we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns the ones we dont know we dont know.
>
> *Donald Rumsfeld, February 2002*

Politics aside, the statement does point to a truth about what we can prepare ourselves for. A security architecture should obviously have measures for handling threats and attacks we know about. These types of threats and attacks are so predictable that we can plan for them with pro-active baseline security measures.

There are also those unknowns that we know may be there. These are foreseeable and while we do not exactly know how they may materialize, we know enough to plan for handling them. Pro-active baseline security will still play a role, but we also need adaptability and must rely more on re-active schemes. These will require detect-and-respond capabilities.

The "unknown unknowns" are a tougher lot to handle. Part of the problem is that we do not know what to look for. Strategies depending on *detection* as a key element will have to be very flexible in order to catch these type of threats. Humans are generally better at recognizing novel threats than automated systems are, and so putting humans in the *detection* loop seems prudent practice. Still, our ability to distinguish between chaotic data and random data can be questioned, and this will surely impact our ability to detect true patterns. Here, automated systems will need to play an important part. Humans are also notoriously slow when compared to machines, and so an attack could be executed and completed well before a human would be able to respond. All-out attacks will of course be detectable, but by then it may be too late.

Digital one-off pinpoint attacks may also be virtually undetectable in the sense that whatever pattern there were would drown out in the noise of normal behaviour. However, these attacks would not normally constitute an attack on the overall system, and may as such be tolerable (from a system point of view).

N. Guaranteed Eventual Failure

Empires come and go. The Roman Empire fell. The British Empire fell. The Soviet Union fell. They seem all to fall in the long term.

Big corporations come and go. If one investigate the destiny of Fortune 500 companies, it is apparent that even large corporation come and go with a fairly high frequency. A comparison of the destiny of Fortune 500 firms in 1955 vs. 2011 shows that 87 percent are gone (from the list) [69]. Furthermore, the life expectancy of Fortune 500 companies have declined from 75 years and down to less than 15 years.

We have little reason to assume that large-scale ICT system will endure indefinitely. This implies that the system will somehow fail. So, as a postulate, we shall claim that all system will eventually fail. It will be interesting to learn if such failure will be related to how the security architecture fares. We know that the weaknesses of the security in the 1G cellular systems was a contributing factor in the demise of those technologies, but it also clear that GSM and other 2G technologies would have replaced the 1G system irrespective of the merits of the security architecture.

What we do not know very much about is how these systems will fail. As long as the systems are critical ICT infrastructures, it may seem that they will either endure or fail in a disruptive collapse. Of course, as technology progresses, the failure may simply be a slow, but probably accelerating, decline into obsoleteness.

VII. Lessons Learned

A. Assets, Nodes, Entities, Threats and Intruders

Make sure that one has an updated inventory of a system assets, the network elements and nodes, the participating parties/entities,the threats and the most likely would-be intruders.

This is a detaining task and it must be done regularly. We advocate using tools for this purpose, and the Microsoft Threat Modeling Tool is a good practical alternative [70].

B. Requirements and Policies

Threat modeling alone does not solve our problems, but it may help significantly in identifying the consistent parts of the security architecture. The threat modeling tool mention above will, for instance, effortlessly allow requirements to be distilled and appropriate protection to be proposed. Threat modelling may, for instance, also make the requirements clearer and it may also help in defining the security policies.

C. Verify Assumptions

One must verify assumption about the system and the security periodically or when there are substantial changes to the system. That is, an audit is called for to verify assumptions about the assets, the principal entities, trust relationships, etc.

Security policies must be adapted according to changes to the assumptions. This is a process oriented task that must take place both for the design phase and for the deployed system(s).

We want to highlight that even non-technical aspects such as trust must be carefully reviewed. We also want to point out the difference between trust and trustworthy. The fact that one trust someone's intentions does not imply that one can trust their ability to behave according to intention. Thus, we must assure ourselves that our partners are trustworthy. The trust assumption should of course be explicit and concrete.

D. Rock Solid Bootstrapping Security

There needs to be a rock solid fundament that will be secure for the foreseeable future. The smart-card has served this purpose in the 3GPP systems on the subscriber side. The smart-card is not tamper-proof, but it has successfully served as a high-trust platform.

That being said, a recently leaked document from the British Government Communications Headquarters (GCHQ), shows that NSA/GCHQ have at least hacked one of the major SIM card manufacturers. The target company, Gemalto, is a large multinational firm based in the Netherlands that produces in the order of 2 billion SIM cards a year. The leak is part of the Snowden files and is published at The Intercept [27]. It is worth to note that Gemalto probably is one of the most security conscious companies out there, but they obviously were not impenetrable in the end. This may in itself be a lesson.

The leak does not weaken our requirement for a rock solid bootstrapping base, but it highlight the need for ensuring that the trust in the base is warranted, enforced and validated.

E. Planned Deprecations

A scalable and evolving system must be able to handle deprecation of almost all cryptographic algorithm, security protocols and security services. The deprecation, needless to say, must be conducted in a secure manner. Backwards compatibility requirements and fallback solutions must be handled in a secure way.

F. Negotiable and Adaptable

Given that one must plan for deprecation of security features/services, one must also plan how to negotiate new features/services. This feature must be built-in and have high assurance. Adaptation may be necessary to account for local requirements, but is vital that adaptations must be fully compliant with a well-defined security policy.

G. Proactive & Reactive Security

Basic security functionality to identify and authenticate principals and entities is necessary, but not sufficient. Adding authorization, protected storage and protect communication is also necessary, but still not sufficient. More may be added, but in the end it is impossible to fully secure the system. This means that one must handle and deal with incidents. Therefore, there is a clear need for intrusion detection and response systems, to deploy firewalls, anti-virus protection, secure backups, secure audit trails etc. The reactive measures must be included in the overall system security plans and subject to revisions as need be.

H. Stability, Resilience and Recovery

System integrity is imperative to ensure a stable and resilient system. System integrity is a system-level characteristic and does not preclude partial or local failures. What is imperative is to prevent the failures to scale. Failures, whether man-made intentional or unintentional, cannot entirely be prevented. Procedures that support mitigation and recovery must be an integral part of the overall system security plan.

I. Configuration Management

Proper planned configuration management, which must include security functionality, is an absolute necessity.

J. Memoryless Security

Security will fail, and then it is prudent that the impact is contained. This speaks strongly in favor of security protocols and crypto systems that are "memoryless". That is, perfect forward secrecy (PFS) should be included as a major principle.

K. Privacy Matters

Privacy is one feature that must be accounted for in all systems that include human users or any kind of data pertaining to humans. This must be planned for from the design phase and handled in all phases of system deployment.

Privacy is, however, also a difficult concept and largely a culturally dependent trait. What can be expect to keep private, and not the least, from whom do we keep information private. Nevertheless, whatever privacy level we decide on, one should ensure that it is credibly maintained.

VIII. Discussion

A. Evolution

In this paper, we have outlined the 3GPP security architecture as it has evolved over more than 25 years. From being an auxiliary service for the few, it has grown to literally cater to billions of subscribers, and the number and types of services provided has changed dramatically over the years. The use-patterns of these systems has changed as well. All in all, there has been a complete transformation of almost all aspects of these systems. During this process, the security architecture has evolved with the system and the changing system context, though not without some noticeable failures and a growing number of security problems.

We have argued that to achieve scalable security architectures that are able to evolve over time, one needs to take into account the fact that almost all assumption one initially had will become false or moot. This means that adaptability and ability to support changes is crucial.

B. Not Fully Justified

The results in this paper cannot be said to be fully supported by the evidence provided in this paper (or in the referenced papers). They results are neither rigorous nor complete. This is to be expected for such a complex issue. Thus, while the results may be valid and true, they will hardly be complete and not always necessary either. That is, the usual "necessary and sufficient" conditions are not really there. Still, experience and empirical evidence should not be discounted, and we advocate that the lessons learned are taken into account, not as mathematical axioms, but inputs to be considered. Therefore, we recommend that scalable evolving security architectures should be designed with these assumption as background.

C. Pervasiveness, Importance and Dependability

This is important in a world where the internet-of-things (IoT) landslide is about to happen and where the systems will be ever more important.

In the wake of the Snowden revelations, it is also clear that cyber-security is under constant pressure, and while we do not want to over-state the Snowden case per se, it should be clear that the cyber-war methods will (over time) become available to many organizations and individuals. Schneier captures this well when he stated that [71]:

> And technology is fundamentally democratizing: today's NSA secret techniques are tomorrow's PhD theses and the following day's cybercrime attack tools.

How stable and durable are our ICT-based future? The internet pioneer Vinton Cerf warns of a "forgotten century", pointing to the risk that the digital material we produce today may be unreadable by tomorrow equipment [72]. He calls out for "digital vellum" to solve this problem. The risk is no less dire for other reasons for ICT collapse, including Black Swan type of massive security failures.

Finally, it is clear that large-scale ICT infrastructures are highly complex and interdependent entities. The security architectures are no less complex. To name a few, we have issues with backwards compatibility and deprecation of old features, issues with migration towards new functionality, issues with integration with other system, ever-changing threats and ever-changing population of users, etc. In short, it is staggeringly complex, and while little of the complexity is the of the "necessarily complex" type, it is not easily reducible either.

D. Forward Directions

In some sense we find ourselves in the same situation as the old Norse Allfather god Odin. He, with his brothers, had created the world, but it turned out he did not understand his own creation. Odin had to sacrifice one eye to drink from the wisdom well Mimir to gain knowledge.

Our ICT systems are highly complex, relatively fragile and strangely resilient at the same time. We also know that there is a cyber security battlefield and we know we are getting ever more dependent on our systems. So, preferably without too much sacrifice, we urgently need to learn more about what works and what does not work in the protection of our critical ICT infrastructures.

There are obviously technical aspects that needs to be studied further, but this is not enough. We also need a better understanding of the societal aspects of security architecture evolution in large-scale critical ICT system.

IX. Conclusion

In this paper, we have investigated the security architectures or the 3GPP systems. Section II is devoted to this topic. In Section III, we focused on how the security architecture must evolve with the systems and a number of aspects that must be considered. Evolution implies changes, and we have also taken a look at some of the reasons one may want or may have to change the security architecture. This is captured in Section IV. In Section V, we presented a whole range of assumptions about the target systems, the security components and the cryptographic primitives. They are not important individually, but we have postulated them as a means to set up a grand picture of what one must keep in mind regarding a large and long-lived critical infrastructure system.

Somewhat loosely inspired by the article "Why Cryptosystems Fail" [32], we provided a whole section of indirect reasons why things may fail. This is captured in Section VI. The arguments and cases presented here do not constitute evidence or proof. We believed, however, that an awareness of these cases and phenomena may be useful for the mindset needed for designing evolving security architectures. In Section VII, we have tried to distill some of the lessons learned from the 3GPP systems. It is by nature incomplete, but may serve as a starting point for a principles and guidelines document for security architecture design. This section contain discussions, but not all aspects are covered in full. In Section VIII, we include a brief discussion of remaining matters.

Overall, our method has largely been a descriptive one, and a deeper theory of security architecture evolution is still missing. One reason for this is that one in all likelihood cannot fully understand this type of system evolution in terms of security methodologies alone.

The lesson learned, it is hoped, should not be isolated to the 3GPP systems, but be applicable to any system of similar magnitude and scope.

References

[1] G. M. Køien, "Challenges for Evolving Large-Scale Security Architectures," in Proceedings of the Eighth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2014), Novemer 16-20, 2014, Lisbon, Portugal. IARIA, 2014, pp. 173–179, ISBN: 978-1-61208-376-6, ISSN: 2162-2116.

[2] 3GPP, TS 43.020, "Security related network functions," 3GPP, France, TS 43.020 (2G), 2014.

[3] 3GPP, TS 21.133, "3G security; Security threats and requirements," 3GPP, France, TS 21.133 (3G), 2001.

[4] 3GPP, TS 33.120, "Security Objectives and Principles," 3GPP, France, TS 33.120 (3G), 2001.

[5] 3GPP, TS 33.210, "3G security; Network Domain Security (NDS); IP network layer security," 3GPP, France, TS 33.210 (NDS/IP), 2012.

[6] 3GPP, TS 33.401, "3GPP System Architecture Evolution (SAE); Security architecture," 3GPP, France, TS 33.401 (3G), 2014.

[7] AVANTSSAR project, "Aslan++ specification and tutorial," FP7-ICT-2007-1, Deliverable 2.3 (update), 01 2008, Available: http://www.avantssar.eu/ [accessed: 2015-06-01].

[8] AVISPA project, Automated Validation of Internet Security Protocols and Applications (AVISPA); AVISPA v1.1 User Manual, 1st ed., AVISPA IST-2001-39252, 06 2006, Available: http://www.avispa-project.org [accessed: 2015-06-01].

[9] N. P. Smart, V. Rijmen, M. Stam, B. Warinschi, and G. Watson, "Study on cryptographic protocols," ENISA, Report TP-06-14-085-EN-N, 11 2014.

[10] European Parliament/European Council, "Directive 2009/136/EC of the European Parliament and of the Council of 25 November 2009 amending Directive 2002/22/EC on universal service and users rights relating to electronic communications networks and services, Directive 2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector and Regulation (EC) No 2006/2004 on cooperation between national authorities responsible for the enforcement of consumer protection laws." EU, Directive 09/136/EC, 2009.

[11] European Council, "European Council Resolution January 1995 JAI 42 Rev 28197/2/95 (Official Journal reference 96C 329/01 4 November 1996)," EU, Resolution, 1995.

[12] European Parliament/European Council, "Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending Directive 2002/58/EC," EU, Directive 06/24/EC, 2006.

[13] J. R. Rao, P. Rohatgi, H. Scherzer, and S. Tinguely, "Partitioning attacks: or how to rapidly clone some gsm cards," in Proceedings of IEEE Symposium on Security and Privacy (2002). IEEE, 2002, pp. 31–41.

[14] M. Kalenderi, D. Pnevmatikatos, I. Papaefstathiou, and C. Manifavas, "Breaking the gsm a5/1 cryptography algorithm with rainbow tables and high-end fpgas," in Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on. IEEE, 2012, pp. 747–753.

[15] P. Papantonakis, D. Pnevmatikatos, I. Papaefstathiou, and C. Manifavas, "Fast, fpga-based rainbow table creation for attacking encrypted mobile communications," in Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on. IEEE, 2013, pp. 1–6.

[16] G. M. Køien, Entity authentication and personal privacy in future cellular systems. River Publishers, 2009, vol. 2.

[17] F. van den Broek, B. Hond, and A. Cedillo Torres, "Security Testing of GSM Implementations," in Engineering Secure Software and Systems, ser. Lecture Notes in Computer Science, J. Jürjens, F. Piessens, and N. Bielova, Eds. Springer International Publishing, 2014, vol. 8364, pp. 179–195.

[18] G. M. Køien, "An introduction to access security in UMTS," Wireless Communications, IEEE, vol. 11, no. 1, Feb 2004, pp. 8–18.

[19] V. Niemi and K. Nyberg, UMTS Security. John Wiley & Sons, 2003.

[20] 3GPP, TS 33.102, "3G Security; Security architecture," 3GPP, France, TS 33.102 (3G), 2014.

[21] 3GPP, TS 33.310, "Network Domain Security (NDS); Authentication Framework (AF)," 3GPP, France, TS 33.310 (NDS/AF), 2014.

[22] G. M. Køien, "Privacy enhanced cellular access security," in Proceedings of the 4th ACM workshop on Wireless security. ACM, 2005, pp. 57–66.

[23] N. Golde, K. Redon, and J.-P. Seifert, "Let me answer that for you: Exploiting broadcast information in cellular networks," in Proceedings of the 22nd USENIX conference on Security. USENIX Association, 2013, pp. 33–48.

[24] N. Gobbo, F. Palmieri, A. Castiglione, M. Migliardi, and A. Merlo, "A denial of service attack to UMTS networks using SIM-less devices," IEEE Transactions on Dependable and Secure Computing, 2014, p. 1.

[25] D. Forsberg, G. Horn, W.-D. Moeller, and V. Niemi, LTE security. John Wiley & Sons, 2012, vol. 1.

[26] A. Gupta, T. Verma, S. Bali, and S. Kaul, "Detecting MS initiated signaling DDoS attacks in 3G/4G wireless networks," in Communication Systems and Networks (COMSNETS), 2013 Fifth International Conference on. IEEE, 2013, pp. 1–60.

[27] J. Scahill and J. Begley, "How spies stole the keys to the encryption castle," The Intercept, 2 2015, Available: https://firstlook.org/theintercept/2015/02/19/great-sim-heist/ [accessed: 2015-06-01].

[28] Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer, and V. Paxson, "The matter of heartbleed," in Proceedings of the 2014 Conference on Internet Measurement Conference, ser. IMC '14. New York, NY, USA: ACM, 2014, pp. 475–488.

[29] Y. Huang and A. K. Ghosh, "Introducing diversity and uncertainty to create moving attack surfaces for web services," in Moving Target Defense, ser. Advances in Information Security, S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, Eds. Springer New York, 2011, vol. 54, pp. 131–151.

[30] M. M. Williamson and J. Léveillé, "An epidemiological model of virus spread and cleanup," Information Infrastructure Laboratory, HP Laboratories Bristol, vol. 27, 2003.

[31] L. Carroll, Through the looking glass: And what Alice found there. Rand, McNally, 1917.

[32] R. Anderson, "Why cryptosystems fail," in Proceedings of the 1st ACM Conference on Computer and Communications Security. ACM, 1993, pp. 215–227.

[33] D. Florêncio and C. Herley, "Where do all the attacks go?" in Economics of Information Security and Privacy III. Springer, 2013, pp. 13–33.

[34] 3GPP, TS 35.202, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification (Release 12)," 3GPP, France, TS 35.202, 2014.

[35] 3GPP, TR 33.908, "3G Security; General report on the design, specification and evaluation of 3GPP standard confidentiality and integrity algorithms," 3GPP, France, TR 33.908, 2001.

[36] 3GPP, TS 35.216, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2; Document 2: SNOW 3G specification (Release 12)," 3GPP, France, TS 35.216, 2014.

[37] D. Dolev and A. C. Yao, "On the Security of Public-Key Protocols," IEEE Transactions on Information Theory, vol. 29, no. 2, 3 1983, pp. 198–208.

[38] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks," Communications Surveys & Tutorials, IEEE, vol. 15, no. 4, 2013, pp. 2046–2069.

[39] I. Jacobs and N. Walsh, "Architecture of the world wide web, volume one," World Wide Web Consortium (W3C), W3C Recommendation, 12 2004.

[40] R. Anderson, Security engineering. John Wiley & Sons, 2008.

[41] G. M. Køien, "Privacy enhanced mutual authentication in LTE," in Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on. IEEE, 2013, pp. 614–621.

[42] ——, "Mutual entity authentication for LTE," in Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International. IEEE, 2011, pp. 689–694.

[43] European Parliament/European Council, “Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.” EU, Directive 95/46/EC, 1995.

[44] M. Mccauley, The Rise and Fall of the Soviet Union, ser. Longman History of Russia. Taylor & Francis, 2014.

[45] R. Swedberg, “The structure of confidence and the collapse of Lehman Brothers,” Research in the Sociology of Organizations, vol. 30, 2010, pp. 71–114.

[46] J. C. Dvorak, “AltaVista, the Biggest Fail Ever,” PC Mag, 07 2013.

[47] N. N. Taleb, Fooled by Randomness: The Hidden Role of Chance in Life and in the Markets. Random House Publishing Group, New York, 2001.

[48] ——, The Black Swan: The Impact of the Highly Improbable. Random House Publishing Group, New York, 2007.

[49] F. van den Broek, “Eavesdropping on GSM: state-of-affairs,” CoRR, vol. abs/1101.0552, 2011, 5th Benelux Workshop on Information and System Security (WISSec 2010), November 2010.

[50] A. AtKisson, Believing Cassandra: An optimist looks at a pessimist’s world. Chelsea Green Publishing Company, 1999.

[51] J. Kruger and D. Dunning, “Unskilled and Unaware of It: How Difficulties in Recognizing One’s Own Incompetence Lead to Inflated Self-Assessments.” Journal of personality and social psychology, vol. 77, no. 6, 1999, p. 1121.

[52] C. N. Parkinson and O. Lancaster, Parkinson’s law: The pursuit of progress. Readers Union [in association with] John Murray, 1959.

[53] P.-H. Kamp, “Why Should I Care What Color the Bikeshed Is?” Frequently Asked Questions for FreeBSD 7. X, 8. X, and 9. X. FreeBSD, 1999, Available: http://bikeshed.com/ [accessed: 2015-06-01].

[54] A. Melnikov, “Rerun: Elliptic curves - preferred curves around 256bit work factor,” CFRG mailing list, 2 2015, Available: http://www.ietf.org/mail-archive/web/cfrg/current/msg06331.html [accessed: 2015-06-01].

[55] H. Berghel, “Cyber Chutzpah: The Sony Hack and the Celebration of Hyperbole,” IEEE Computer magazine, vol. 48, 02 2015, pp. 77–80.

[56] B. Schneier, “Beyond fear,” Copernicus Book, New York, 2003.

[57] ——, “Reconceptualizing Security,” in LISA’08: 22nd Large Installation System Administration Conf, 2008.

[58] R. P. Feynman, “Cargo cult science,” in Surely You’re Joking, Mr. Feynman, 1st ed. W. W. Norton, 1985, Originally a 1974 Caltech commencement address.

[59] G. Hardin, “The Tragedy of the Commons,” science, vol. 162, no. 3859, 1968, pp. 1243–1248.

[60] B. Schneier, Liars and outliers: enabling the trust that society needs to thrive. John Wiley & Sons, 2012.

[61] D. Adams, Life, the Universe and Everything: Hitchhiker’s Guide 3. Tor UK, 1984, vol. 3.

[62] B. Latané and J. M. Darley, “Bystander Apathy,” American Scientist, 1969, pp. 244–268.

[63] J. M. Darley and B. Latane, “Bystander intervention in emergencies: diffusion of responsibility.” Journal of personality and social psychology, vol. 8, no. 4p1, 1968, p. 377.

[64] M. Pease, R. Shostak, and L. Lamport, “Reaching Agreement in the Presence of Faults,” Journal of the ACM (JACM), vol. 27, no. 2, 1980, pp. 228–234.

[65] L. Lamport, R. Shostak, and M. Pease, “The Byzantine Generals Problem,” ACM Transactions on Programming Languages and Systems (TOPLAS), vol. 4, no. 3, 1982, pp. 382–401.

[66] P. Cvitanović, R. Artuso, R. Mainieri, G. Tanner, and G. Vattay, Chaos: Classical and Quantum. Copenhagen: Niels Bohr Institute, 2012, Available: http://chaosbook.org/ [accessed: 2015-06-01].

[67] M. Scheffer, S. R. Carpenter, T. M. Lenton, J. Bascompte, W. Brock, V. Dakos, J. Van De Koppel, I. A. Van De Leemput, S. A. Levin, E. H. Van Nes et al., “Anticipating critical transitions,” science, vol. 338, no. 6105, 2012, pp. 344–348.

[68] A. J. Veraart, E. J. Faassen, V. Dakos, E. H. van Nes, M. Lürling, and M. Scheffer, “Recovery rates reflect distance to a tipping point in a living system,” Nature, vol. 481, no. 7381, 2012, pp. 357–359.

[69] J. Chew, “Fortune 500 Extinction,” csinvesting.org, 01 2012, Available: http://csinvesting.org/2012/01/06/fortune-500-extinction/ [accessed: 2015-06-01].

[70] A. Shostack, Threat modeling: Designing for security. John Wiley & Sons, 2014.

[71] B. Schneier, “Why the NSA’s Attacks on the Internet Must Be Made Public,” The Guardian.com, 10 2013, Available: http://www.theguardian.com/commentisfree/2013/oct/04/nsa-attacks-internet-bruce-schneier [accessed: 2015-06-01].

[72] I. Sample, “Google boss warns of ’forgotten century’ with email and photos at risk,” The Guardian.com, 2 2015, Available: http://www.theguardian.com/technology/2015/feb/13/google-boss-warns-forgotten-century-email-photos-vint-cerf [accessed: 2015-06-01].