BigMac 🍔: Performance Overhead of User Plane Integrity Protection in 5G networks

Thijs Heijligenberg
Radboud University
Nijmegen, Netherlands
theijligenberg@cs.ru.nl

Guido Knips
Radboud University
Nijmegen, Netherlands
gknips@cs.ru.nl

Christian Böhm
Ruhr University
Bochum, Germany
christian.boehm@rub.de

David Rupprecht
Radix Security
Germany, Bochum
david@radix-security.com

Katharina Kohls
Radboud University
Nijmegen, Netherlands
kkohls@cs.ru.nl

ABSTRACT

5G introduces a series of new security features that overcome known issues of the previous mobile generations. One of these features is integrity protection for user plane data. While this addition protects against manipulations like DNS spoofing, it also introduces extra overhead to user plane traffic. As it is optional to enable, this additional overhead can be the decision point for network operators to avoid the additional security feature. In this work, we investigate the overhead induced by different integrity protection algorithms and test the burden they add to the workload of a device. Our results indicate how visible performance differences would be on the end-devices of users, and how the performance of the algorithms differs in isolation. With these results we aim to initiate a discussion regarding the benefits of enabling user plane integrity protection and to overcome misconceptions regarding the performance impairments for end users.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**; *Security protocols.*

KEYWORDS

5G, integrity protection, latency, performance

ACM Reference Format:
Thijs Heijligenberg, Guido Knips, Christian Böhm, David Rupprecht, and Katharina Kohls. 2023. BigMac 🍔: Performance Overhead of User Plane Integrity Protection in 5G networks. In *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '23), May 29-June 1, 2023, Guildford, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3558482.3581777

1 INTRODUCTION

Mobile networks are ubiquitous in our society [14] and, throughout the years, different mobile generations gradually increased the security and performance of our networks. When switching from our current mobile generation, 4G, to the upcoming 5G technology, a series of fundamental and security-relevant changes are part of the transition process. One example of this is the introduction of User Plane Integrity Protection (UP IP), which overcomes critical security flaws in our current 4G networks [19, 20]. With Standalone (SA) 5G, i. e., 3rd Generation Partnership Project (3GPP) Release 16, UP IP becomes *mandatory* to implement.

Although this is a promising step forward, the 3GPP only specifies the implementation as mandatory and leaves it *optional* to use for network operators. This leads to a situation where all 5G SA equipment adhering to 3GPP Release 16 has the capability to use UP IP. However, operators can still decide whether UP IP is actually enabled for a specific connection. As the additional security layer also introduces additional processing steps, connections might face a visible overhead [14].

Ongoing discussions between operators and vendors indicate that the main concerns regarding UP IP involve latency, throughput, and power consumption [1]. Such performance impairments jeopardize critical sales criteria of 5G while the existing security threats of missing integrity protection are not transparent to consumers. Besides these strategic assumptions, the *practical* implications of UP IP remain unclear. Since mobile devices use different baseband chips with fundamental architectural differences, the performance impairments might differ across the various existing implementations. The varying hardware setups of base stations complicate this missing evaluation step even further [10].

In this work, we aim for a first comparative performance evaluation of the established algorithms for UP IP. To this end, we provide empirical measurement results and discuss how our observations can be used to indicate the overhead of enabled integrity protection. In our experimental evaluation we focus on two different aspects related to the performance of UP IP. First, we analyze specified standard algorithms in isolation to document their conceptual differences and different influencing factors for their overhead. Second, we conduct a practical evaluation that focuses on the performance impairments that are seen on the devices of end-users. Our experiments not only give an indication of the algorithmic overhead of integrity protection in the current standard, but they also reveal network behavior that masks potential performance impairments.

Our results indicate significant performance differences within the current state of the art of mobile network integrity protection algorithms when run in isolation. In an actual mobile network

WiSec '23, May 29-June 1, 2023, Guildford, United Kingdom

ACM ISBN 978-1-4503-9859-6/23/05.
https://doi.org/10.1145/3558482.3581777

connection however, the performance difference between the algorithms is negligible for both latency and throughput. Disabling UP IP altogether does result in a small but notable decrease in latency and a more stable throughput. In our work, we make the following contributions:

- We evaluate the runtime and throughput of algorithms used in mobile networks and suggest possible improvements or alternatives for future implementations and the 3GPP specification.
- We evaluate latency and throughput of the connection between a User Equipment (UE) and a mobile network with and without UP IP and for different integrity protection algorithms.

2 BACKGROUND

At the beginning of each connection, the phone and network perform an Authentication and Key Agreement (AKA) procedure to establish the prerequisites for a secure transmission channel. From this point on, it is possible to encrypt and integrity protect transmissions. While encryption is applied to both control and user plane data, the application of integrity protection is only mandatory for the control plane.

2.1 PDCP security

The Packet Data Convergence Protocol (PDCP) layer is responsible for the encryption and integrity protection and has a maximum transfer unit of 8188 bytes. It uses a MAC-then-Encrypt scheme which calculates the MAC logically before the ciphering [3]. To satisfy different quality requirements, user traffic is further divided into different dedicated bearers that serve as logical channels. The bearers provide a default channel for Internet service, and two bearers for Voice over LTE (VoLTE) traffic. It is possible to assign individual parameters to a bearer, e. g., cryptographic algorithms [2].

2.2 Security Algorithms

5G and LTE share the same set of security algorithms although their naming differs. Throughout the paper we refer to the 5G nomenclature. The encryption algorithms are named NEAx; integrity algorithms are named NIAx where in both cases $x = 1, 2, 3$.

To analyze the overhead of UP IP, we mainly focus on the integrity procedures NIA1, NIA2, and NIA3. Only NIA1 and NIA2 must be implemented by both the user and the network [4], while NIA3 is optional. All algorithms share the same basic interface and use individual concepts of underlying base algorithms. All algorithms have a linear time complexity and either a constant or linear space complexity [12].Currently the following algorithms are specified for the purpose of protecting traffic:

- NEA1/NIA1: SNOW3G
- NEA2/NIA2: AES (CTR mode/CMAC mode)
- NEA3/NIA3: ZUC

2.2.1 Algorithm: NIA1. NIA1 uses the stream cipher SNOW3G and was originally specified for the use in 3G systems in 2006 [11] under the name UEA2. The message input has a maximum length of 2^{32} bits and is divided into 64-bit chunks. SNOW3G is used to compute a keystream of five 32-bit words. This keystream is used to calculate the MAC tag, making use of mapping functions in $GF(2^{64})$.

2.2.2 Algorithm: NIA2. NIA2 makes use of AES in CMAC mode to calculate the MAC tag [9]. In comparison to NIA1 there is no such thing as a keystream that can be generated beforehand, thus, all calculations take place on-the-fly. The sequence number, bearer identity, and direction are appended to the message and serve as input for the AES-CMAC algorithm.

2.2.3 Algorithm: NIA3. NIA3 is the second algorithm to utilize a keystream for building the MAC, and is based on the stream cipher ZUC. It generates a stream that is 64 bits longer than the input message. The MAC tag is then calculated via XOR operations.

2.3 Frame structure

Mobile networks are timed using *frames*, which are always 10 milliseconds long. These consist of *subframes* which are 1 millisecond long. The frame structure dictates how soon a message presented by higher layers can be sent, which in turn translates to latency to the end user. Subframes are dedicated to either uplink or downlink traffic. There are two ways to divide between traffic directions:
Frequency division duplex (FDD): there are two distinct frequencies used for uplink and downlink. This means that there is always an uplink or downlink subframe available on short notice, which is useful for lowering traffic latency. The downside is that this uses a more substantial part of the spectrum, and that the amounts of downlink and uplink capacities are always equal.
Time division duplex (TDD): the subframes are uplink or downlink based on their index, and this structure is configured by the network. This allows for asymmetric allocation of resources and more flexible design. The downside is that the timing constraints on devices are harder to meet, and that there needs to be some buffer time allocated between the directions.

There is the option to use more elaborate designs using multiple carriers, but a basic setup will utilize one of these two options. The use of FDD or TDD is linked to the frequency band; the specification provides a list of bands and their duplex forms.

3 ALGORITHM EVALUATION

As a first step in the analysis procedure, we focus on the performance of security algorithms in a reference implementation. This allows us to compare conceptual differences of the algorithms regarding their latency and throughput aspects. Power consumption, which is the third identified drawback of UP IP (see § 1), is left for future work. Our initial experiments focus on the basic concept of these algorithms in isolation. There are multiple options to increase performance in both latency and throughput (§ 5).

3.1 Experimental Setup

For all experiments, we use existing C/C++ implementations of the different NIA/NEA algorithms as implemented in the srsRAN suite [13]. These implementations refer to the official specification of NIA1 and NIA3 and serve as a reference. For the sake of comparability, we do not use any hardware acceleration for the AES-based algorithms (NIA2/NEA2), but instead use a reference

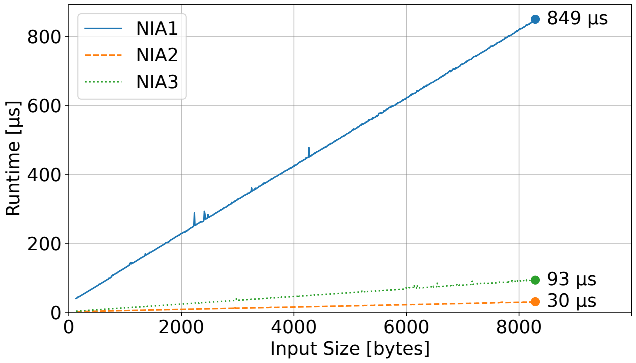

Figure 1: Runtime of integrity protection algorithms in a performance benchmark.

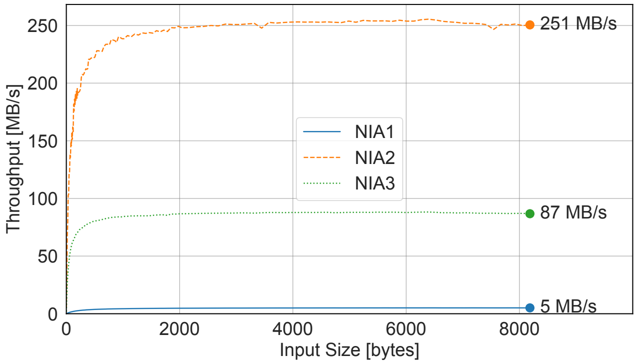

Figure 2: Throughput for integrity protection algorithms in a performance benchmark.

software implementation. This is different from the default srsRAN implementation, which uses the `mbedtls` library.

We perform all experiments on an AMD Ryzen 7 PRO 4750U. Each experiment runs on a bounded core to guarantee stability. We measure the runtime by calculating the difference between start and end time. For the throughput, we run the algorithm for 5 seconds and measure the total amount of data that was integrity protected. We vary the input length throughout the experiments to show the effects of different sizes on runtime and throughput.

3.2 Latency

The time it takes to run an integrity protection algorithm depends on the size of the input. In our case the size is limited to the length of a PDCP frame, which is 8188 bytes, see § 2.1. We present these results in Figure 1.

Note that these results represent a reference implementation, and therefore solely focus on the integrity protection running time. In deployed systems, additional influencing factors can affect the overall latency of a connection. The way this measured latency carries over to a real system is measured in § 4.3.1.

Furthermore, the latency induced by UP IP is directly correlated with the running time of the selected cryptographic algorithm. There are, however, some possible techniques to get around this hard limitation. Running time of cryptographic primitives is a well-studied area, which also provides some more optimized implementations which can greatly reduce this side effect of UP IP (see § 5).

3.3 Data throughput

One of the key promises of 5G is data throughput [14]. This is to some extent proportional to latency, but due to the different designs of the algorithms this does not directly relate. We provide results for this factor in Figure 2.

Throughput on a general-purpose system is, similar to the case of latency, determined by a large number of factors. However, throughput is not additive like latency, and UP IP may turn out to not be a bottleneck. This may lead to the practical effect of UP IP on throughput not being symmetrical with the effect on latency. We will study this in § 4.3.2.

3.4 Conclusion

The three algorithms differ greatly with respect to both latency and throughput in an isolated environment. The measured latency is well below normal operational latency of a mobile network and may therefore be of little impact. The throughput capacity is not likely to be a bottleneck for NEA2 and NEA3, while NEA1 may have issues in this aspect.

4 PERFORMANCE EVALUATION

In the next analysis step, we focus on the performance of the integrity protection algorithms in a real-world scenario. This provides us with insights about their performance within a complex system and the consequences that would be visible to end-users. We focus on latency and throughput, which are regarded as two of the key side effects of UP IP. The third and last key side effect of UP IP is power consumption. In a set of preliminary experiments we observed that we cannot reliably distinguish the effect of UP IP from other influencing factors that affect the power consumption of a Commercial Off-The-Shelf (COTS) UE. Such factors involve the physical transmissions, the background activities of the operating system, and the general battery characteristics of a device. We leave a more elaborate analysis of the power consumption of UP IP to future work.

4.1 Experimental Setup

Our setup consists of an Amarisoft Callbox Classic acting as both a base station and core network. This system offers an off-the-shelf standalone 5G implementation of commercial quality. The devices we use as UE are:

- Oneplus 10 5G
- Realme GT NEO 3

Both phones have a 5G Release 16-compliant baseband. We ran our experiments primarily on the Oneplus 10. We use the Realme GT NEO 3 for verification of a subset of the results, as it does not allow the full range of configurations and setups needed in our experiments. During the measurements, we keep the phones in close proximity and line-of-sight conditions to the base station and avoid any user activity on the phone during testing.

4.1.1 Latency. For latency measurements, we use the `ping` command, which measures roundtrip time of a simple message. Due to the small size of these packets the latency is purely caused by the time it takes to take all steps needed to transmit one message, and not due to throughput limitations.

When measuring the latency of a TDD network, there is latency induced by the time it takes before a suitable uplink or downlink slot is available. Consequently, the arrival times of packets can only occur at specific times during a frame (see § 2.3). When the ping interval aligns with frame timing, some synchronization patterns occur. To avoid this effect, we use a message interval of 107 milliseconds which circumvents the pattern.

4.1.2 Throughput. For the throughput experiments, we use the `curl` program with output directed to a sink. In addition to the controlled traffic that we generate in our experiments, there is an additional amount of background traffic caused by the operating system of the phone. Although these extra transmissions are not part of the controlled traffic, we can assume constant rates for all repetitions of the experiments. Consequently, the system traffic can be considered as negligible noise to the throughput measurements.

As a data source we use a simple `HTTP` server that sends an infinite stream of data in the same local network as the Callbox. To avoid any scheduling effects and an overload situation at the mobile network, we assure that the throughput rates of our measurements stay below the processing limits of the Callbox. Consequently, all observed limitations result from effects at the air interface.

4.2 Configuration

In all scenarios we configured the network to either not use UP IP, or to require UP IP and only allow one specific integrity protection algorithm. The phones are able to use all available algorithms.

4.2.1 Latency. For these experiments we use the preconfigured standalone 5G configuration of the Amarisoft Callbox. This TDD configuration gives a data rate and latency which are comparable with a commercial network, see § 4.4. We repeat the same experiments in a dedicated low-latency TDD configuration. This allows us to eliminate effects introduced by the network setup rather than the UP IP setups. All experiments consisted of 10000 messages.

Note that the Callbox also offers FDD configurations. As preliminary experiments reveal an overall higher latency in the measurements we exclude these configurations from our experiments to avoid any setup-related bias.

4.2.2 Throughput. To measure the throughput of connections, we refer to the same configuration as in the latency experiments. All configurations were run for 3000 seconds spread over 5 sessions.

4.3 Results

Within the latency and throughput setups, we observe the following characteristics.

4.3.1 Latency. The results for our latency experiments can be seen in Fig. 3. There is a large difference between the configuration optimized for latency and the standard configuration. For both there is a significant increase in latency between the tests without UP IP and those with the various UP IP algorithms enabled. There is no clear difference between the different UP IP algorithms.

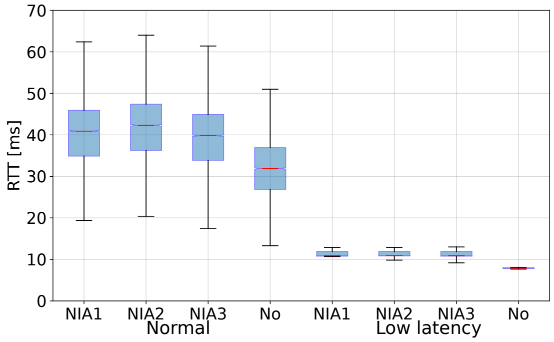

Figure 3: Latency in a practical 5G setup for different configurations as measured by `ping`.

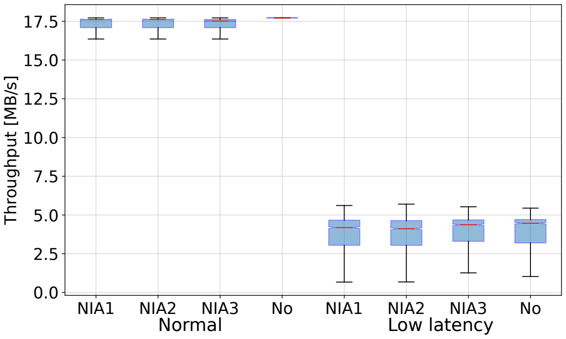

Figure 4: Throughput in a practical 5G setup for different configurations as measured by `curl`.

4.3.2 Throughput. The results for our throughput experiments can be seen in Fig. 4. Again, there is a large difference between the configuration optimized for latency and the standard configuration; in this case the low-latency configuration results in lower throughput. For both there is no significant increase in throughput between the tests without UP IP and those using the various UP IP algorithms, with the only variant standing out being the standard setup without UP IP. This variation might be more optimized for stability. There is no clear difference between the different UP IP algorithms.

4.4 Real-world results

To get a better idea of how representative our lab setup is, we conduct reference experiments with public commercial networks. To this end, we use the same test phone and equip it with a SIM card of a commercial network operator. The network in reach offers non-standalone 5G and uses an FDD band. Our measurements lead to the following latency and throughput observations:

- Latency to gateway: 36 ms (±10 ms)
- Throughput: 18.0 MB/s (±2.5 MB/s)

While this data only covers a single commercial network, it does provide an indication of how close the performance of our setup is to that of a commercial network.

4.5 Conclusion

There is a small difference between enabling UP IP and disabling it, especially in latency. There is no difference between the different algorithms. This indicates that the differences between the algorithms we measured in § 3 can be overcome. There is overhead introduced by UP IP in general, although it is possible that this is an artifact of our setup.

5 POSSIBLE IMPROVEMENTS

While the effect of UP IP on throughput, latency, and power consumption will always be a factor, there are approaches to improve its performance.

5.1 Parallel execution

Modern multi-core architectures allow improvements in performance through parallelization. In our context, where traffic is both encrypted and authenticated, this could be achieved by computing the Message Authentication Code (MAC) in parallel with the ciphertext. Although MAC-then-Encrypt schemes require a MAC of the message before applying the ciphering, we are still able to parallelize with the following procedure: First, we encrypt and compute the MAC independently, which can be done as both algorithms access the plaintext. In contrast to the classical MAC-then-Encrypt sequence, we do not append the MAC to the plaintext before the entire packet is encrypted. Instead, the encryption algorithm puts out additional keystream bytes next to the encrypted message. In a joined step, we encrypt (XOR) the MAC with the keystream bytes and append the encrypted MAC to the encrypted packet. One limitation to this is that this can only be done as a sender as MAC calculation requires a plaintext.

5.2 Phase separation

For the NIA1 and NIA3 algorithms it is possible to generate the MAC keystream before the actual message payload is known. For NIA3 the keystream generation does require the length of the payload, which for some types of traffic can be highly predictable. This technique is especially helpful in contexts where latency is key.

5.3 Architecture

Many modern hardware platforms offer dedicated instructions for certain cryptographic primitives, which allows carrying out procedures in fast hardware implementations. Such a dedicated instruction set exists for NIA2 on many platforms, as it is based on AES, while for NIA1 and NIA3 there only exists academic work [15, 16]. Whether this can be applied depends on the device instruction set.

Another architecture-specific optimization that can be applied is the use of specialized instructions for certain large operations that occur in cryptographic computations. The most prominent example of this is vectorization of arithmetic operations in a loop, which can be automatically applied by modern compilers. Most platforms provide a set of these instructions, but since this is not a unified format any platform-specific optimizations do not carry over directly.

5.4 Alternative algorithms

While our performance results were not constrained by the differences between the algorithms this may not be the case for future applications of 5G, where the limits of latency and throughput are pushed even further. This would motivate the use of new and performance-optimized algorithms such as GCM or Poly1305. However, these raw algorithms do not fit the specification's current interface concept. To overcome this two strategies are available.

One option is to give new algorithms the same individual treatment as the existing ones. For example, integrating Poly1305 into this setting requires generating a one-time key used as input for the integrity algorithm. We can use an encryption algorithm to accomplish this step, which is similar to NIA1. Due to this similarity to an existing algorithm, it is trivial to repeat this step for Poly1305. To realize this in the specification, the 3GPP must agree on an encryption algorithm that provides the one-time key. Options range from the existing SNOW3G, AES, and ZUC algorithms to the newer SNOW-V [10] or ChaCha.

The second option is to take a step back from the current concept to create a closer connection between the encryption and integrity algorithms and their interaction. This step is in line with the AEAD approach that combines suited couples of algorithms, e. g., AES-GCM, ChaCha-Poly1305. A positive side-effect of this is the increased trustworthiness of the combined algorithm. For example, we can assume that ChaCha-Poly1305 is fully secure if the underlying ChaCha encryption is secure. Despite the advantages on different levels, this adjustment requires a more significant conceptual change in the current specification.

6 DISCUSSION

Enabling UP IP does introduce overhead for 5G connections. Our experiments indicate that there is a measurable difference in practice between enabling UP IP and disabling it, while our algorithm analysis shows clear differences between the individual algorithms. We discuss how our practical performance experiments and our isolated algorithm analysis relate. We provide pointers to future work based on the effects we observe throughout our experiments.

6.1 Algorithm results in practice

While our algorithm analysis in § 3 shows a clear difference between the three UP IP algorithms, there is no discernible difference between them in a practical setting. Our performance evaluation does indicate a clear effect of UP IP on latency, but little to no effect on throughput. The latter could be caused by UP IP not being a bottleneck factor for mobile traffic, as mentioned in § 3.3. In all practical cases the effect of UP IP is limited enough to not significantly impact the user experience, especially since latency has less of an effect on most mobile users than throughput [18].

6.2 Implementation

While we have tested reference implementations (§ 3) and discussed possible improvements to these (§ 5), implementation details of the algorithms are opaque to us (§ 4). Authors of future work could gain additional insights into a system with provided source code, although there are currently no open-source frameworks which offer performance of the same quality as commercial networks.

Our current setup produces some unexplained effects that may be due to implementation. One example of this is the fact that all UP IP algorithms gave the same level of latency increase and that this increase is higher in a normal than in a low-latency configuration. Other implementations could also benefit from using a FDD configuration. Low-latency communication is a complicated engineering challenge that could lead to effects around UP IP that are not directly caused by the different algorithms.

6.3 Power consumption

In our current experiments we were unable to extract reliable data on battery consumption in a phone. This is partly caused by incomplete information regarding what the phone regards as “fully charged”. It is also impossible to fully control an off-the-shelf smartphone operating system, which causes noise regarding which background processes are active. Future work would have to get a more complete insight into power management and limit the system to achieve consistent results.

7 RELATED WORK

As the upcoming mobile generation 5G is the first to implement integrity protection for user plane traffic, prior work in this context is limited. However, we can learn from the history of cryptography in previous mobile generations and related work in the context of performance optimizations for algorithms.

Encryption has been a feature of mobile networks since the first widely-available generation 2G (GSM). Prior work shows how these algorithms can be broken [6] or even are weakened by design [7]. The next generation 3G (UMTS) introduces integrity protection for control traffic and stronger cryptographic algorithms based on KASUMI, which is vulnerable to a related key attack [8], and SNOW3G [11], which is still in service today in 4G and 5G. To further improve security standards 4G (LTE) introduces AES and ZUC, which were carried over to 5G.

Over the years there has been research on optimization of the cryptographic procedures used, such as for SNOW3G [15], ZUC [16], and mainly for AES [5, 17]. There has also been research on the complexity of these procedures [12]

8 CONCLUSION

Integrity protection is a fundamental security mechanism for 5G to ensure data integrity and authenticity. Despite these crucial benefits for the security of 5G, integrity protection is *optional* to use. Performance impairments like a higher latency or decreased throughput contradict the high intended performance standards of 5G and are an obstacle to the large-scale use of integrity protection. In this work, we provided a first experimental analysis of the benchmark performance of specified UP IP algorithms, as well as a first evaluation of performance in practice. We discuss how differences between the algorithms do not directly result in performance differences. Practical performance is only impacted significantly in terms of latency, and we point out various implementation details that could lead to performance improvements. This shows that the overhead caused by UP IP does not need to impact users, which paves the way for more secure mobile networks.

ACKNOWLEDGMENTS

This work was in part funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC 2092 CASA - 390781972. Further, it was in part supported by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) under the project ENCRY.2021.001.

REFERENCES

[1] 2020. 3GPP_TSG_RAN@LIST.ETSI.ORG archive, Subject: Re: [Integrity_protection] Final decision on RP-200505.

[2] 3GPP. 2021. *3GPP System Architecture Evolution (SAE); Security architecture.* Technical Specification (TS) 33.401. 3rd Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/33401.htm Version 17.0.0.

[3] 3GPP. 2021. *NR; Packet Data Convergence Protocol (PDCP) specification.* Technical Specification (TS) 38.323. 3rd Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/38323.htm Version 16.6.0.

[4] 3GPP. 2022. *Security architecture and procedures for 5G System.* Technical Specification (TS) 33.501. 3rd Generation Partnership Project (3GPP). http://www.3gpp.org/DynaReport/33501.htm Version 17.4.1.

[5] Nabihah Ahmad, Rezaul Hasan, and Warsuzarina Mat Jubadi. 2010. Design of AES S-Box using combinational logic optimization. In *2010 IEEE Symposium on Industrial Electronics and Applications (ISIEA)*. IEEE, 696–699.

[6] Elad Barkan, Eli Biham, and Nathan Keller. 2003. Instant ciphertext-only cryptanalysis of GSM encrypted communication. In *Annual international cryptology conference*. Springer, 600–616.

[7] Christof Beierle, Patrick Derbez, Gregor Leander, Gaëtan Leurent, Håvard Raddum, Yann Rotella, David Rupprecht, and Lukas Stennes. 2021. Cryptanalysis of the GPRS Encryption Algorithms GEA-1 and GEA-2. Cryptology ePrint Archive, Report 2021/819. https://ia.cr/2021/819.

[8] Eli Biham, Orr Dunkelman, and Nathan Keller. 2005. A Related-Key Rectangle Attack on the Full KASUMI. In *Advances in Cryptology - ASIACRYPT 2005*, Bimal Roy (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 443–461.

[9] Morris J Dworkin et al. 2016. Recommendation for block cipher modes of operation: The CMAC mode for authentication. (2016).

[10] Patrik Ekdahl, Thomas Johansson, Alexander Maximov, and Jing Yang. [n. d.]. A new SNOW stream cipher called SNOW-V. *Transactions on Symmetric Cryptology* ([n. d.]).

[11] ETSI/SAGE. [n. d.]. Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 UIA2. Document 1: UEA2 and UIA2 Specification. https://www.gsma.com/security/wp-content/uploads/2019/05/uea2uia2d1v21.pdf.

[12] El Hajj Ghizlane Orhanou. [n. d.]. The New LTE Cryptographic Algorithms EEA3 and EIA3. http://www.naturalspublishing.com/files/published/zf215q192wiz2s.pdf.

[13] Ismael Gomez-Miguelez, Andres Garcia-Saavedra, Paul D Sutton, Pablo Serrano, Cristina Cano, and Doug J Leith. 2016. srsLTE: an open-source platform for LTE evolution and experimentation. In *ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH)*.

[14] ITU. 2021. Detailed specifications of the terrestrial radio interfaces of International Mobile Telecommunications-2020 M.2150-0.

[15] Paris Kitsos, George Selimis, and Odysseas Koufopavlou. 2008. High performance ASIC implementation of the SNOW 3G Stream Cipher. In *Conference on Very Large Scale Integration (VLSI-SOC '08)*.

[16] Paris Kitsos, Nicolas Sklavos, George Provelengios, and Athanassios N. Skodras. [n. d.]. FPGA-based performance analysis of stream ciphers ZUC, Snow3g, Grain V1, Mickey V2, Trivium and E0. *Microprocessors and Microsystems* ([n. d.]).

[17] Henry Kuo and Ingrid Verbauwhede. 2001. Architectural optimization for a 1.82 Gbits/sec VLSI implementation of the AES Rijndael algorithm. In *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 51–64.

[18] Reiner Ludwig. 2022. Who cares about latency in 5G? https://www.ericsson.com/en/blog/2022/8/who-cares-about-latency-in-5g

[19] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. 2019. Breaking LTE on Layer Two. In *IEEE Symposium on Security & Privacy (SP)*. IEEE.

[20] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. 2020. IMP4GT: IMPersonation Attacks in 4G NeTworks. In *ISOC Network and Distributed System Security Symposium (NDSS)*. ISOC.