A Note on Stream Ciphers that Continuously Use the IV

Matthias Hamann[1], Matthias Krause[1] and Willi Meier[2]

[1] University of Mannheim, Germany, {hamann,krause}@uni-mannheim.de
[2] FH Nordwestschweiz, Switzerland, willi.meier@fhnw.ch

Abstract. Time-memory-data tradeoff (TMD-TO) attacks limit the security level of many classical stream ciphers (like E_0, A5/1, Trivium, Grain) to $n/2$, where n denotes the inner state length of the underlying keystream generator. This implies that to withstand TMD tradeoff attacks, the state size should be at least double the key size. In 2015, Armknecht and Mikhalev introduced a new line of research, which pursues the goal of reducing the inner state size of lightweight stream ciphers below this boundary by deploying a key-dependent state update function in a Grain-like stream cipher. Although their design Sprout was broken soon after publication, it has raised interest in the design principle, and a number of related ciphers have been suggested since, including Plantlet, a follow-up of Sprout, and the cipher Fruit.

In 2017, Hamann et al. showed that the initial hope of achieving full security against TMD-TO attacks by continuously using the secret key has failed. In particular, they demonstrated that there are generic distinguishing attacks against such ciphers with a complexity significantly smaller than that of exhaustive key search. However, by studying the assumptions underlying the applicability of these attacks, they came up with a new design idea for small-state stream ciphers, which is based on also continuously using the public IV as part of the state update. The authors conjectured that this design principle might allow to finally achieve full security against TMD-TO attacks.

In this note, we take their idea one step further. While Hamann et al. aimed for improving the security of small-state stream ciphers that continuously use the secret key against distinguishing, we explain here that also other stream cipher constructions can benefit from continuously using the IV. In particular, our approach allows for thwarting the well-known TMD-TO inner state recovery attacks of Babbage and Biryukov and Shamir without using the secret key more than once.

Keywords: Stream Ciphers · Lightweight Cryptography · Time-Memory-Data Tradeoff Attacks

1 Introduction

Stream ciphers have a long history when it comes to protecting digital communication. In 1987, Ronald L. Rivest designed RC4 [Sch95], which was later used in SSL/TLS [DR08] and the wireless network security protocols WEP [Ins97] and TKIP (often called WPA) [Ins04]. Other well-known stream cipher examples are E_0 of the Bluetooth standard [SIG14] and A5/1 of GSM [BGW99]. Unfortunately, E_0 and A5/1 have been shown to be highly insecure (see, e.g., [LMV05] and [BB06]) and RC4 also shows severe vulnerabilities, which led to its removal from the TLS protocol [Pop15] and rendered other protocols like WEP insecure [FMS01]. In 2004, the eSTREAM project [ECR08] was started in order to identify new stream ciphers for different application profiles. In the hardware category, aiming at devices with restricted resources, three ciphers are still part of the eSTREAM portfolio after the latest revision in 2012: Grain v1 [HJM06], MICKEY 2.0 [BD06] and Trivium [CP05].

Common to these three ciphers is that they have an inner state length of at least twice the size of the targeted security level against key recovery attacks. This is due to the inherent vulnerability of classical stream ciphers (i.e., stream ciphers which compute the keystream based on a so-called *initial state*) against time-memory-data tradeoff (TMD-TO) attacks like those of Babbage [Bab95] and Biryukov and Shamir [BS00], which allow to recover some inner state during keystream generation (and, usually, also the corresponding initial state by clocking the cipher backwards) with an overall attack complexity of $2^{n/2}$, where n denotes the inner state length of the underlying keystream generator (KSG). If the state initialization algorithm, which computes the initial state from a given key/IV pair, is efficiently invertible (as it is, e.g., for Trivium and Grain), knowing the initial state immediately reveals the secret key. And even if the state initialization algorithm is not efficiently invertible, variants of such TMD-TO attacks can allow for key recovery, e.g., by targeting the inner state at $t = 0$, which often contains the secret key (cf. Trivium and Grain). A generic view on these attacks is provided in [HK15] along with a corresponding complexity analysis.

In 2015, a new line of research emerged with the publication of Sprout [AM15] by Armknecht and Mikhalev, which pursues the goal of reducing the size of the volatile inner state of lightweight stream ciphers below this magic boundary formerly induced by TMD-TO attacks. We will refer to such ciphers, whose volatile inner state size is less than twice the key size, by the term *small-state stream ciphers.* Sprout has a Grain-like structure and uses two 40-bit feedback shift registers. Compared to conventional stream ciphers like Grain v1, the characteristic difference of Sprout is that the 80-bit key is not only accessed during the state initialization but also continuously used as part of the state update during the subsequent keystream generation phase. Even though Sprout was broken shortly after publication (see, e.g., [LNP15], [ZG15], [Ban15], [EK16]), it has sparked interest in the underlying design principle and related ciphers like Plantlet [MAM17] and Fruit [GHX16] have been suggested since.

In [HKMZ17], Hamann et al. give an overview over this new class of *small-state stream ciphers*, which, at the moment, consists of the three Sprout-like ciphers (i.e., ciphers that continuously use the secret key) Sprout, Plantlet, and Fruit as well as the stream cipher Lizard [HKM17], which uses a different, non-Sprout-like approach to achieve increased security against TMD-TO *key recovery* attacks. Furthermore, the authors of [HKMZ17] present a generic distinguisher against Sprout-like ciphers with a complexity significantly smaller than that of exhaustive key search. To thwart this kind of attack, Hamann et al. suggest that these ciphers should not only continuously use the secret key but also continuously use the public IV in their state update. However, for other, non-Sprout-like stream ciphers, they conclude:

> Other small-state ciphers like Lizard [27] would hardly benefit for the following reason: If the secret key is not used after the state initialization, there is always the possibility of a TMD tradeoff inner state recovery attack like those by Babbage [3] or Biryukov and Shamir [10]. Such an attack will have complexity half the size of the volatile inner state, independent of whether the IV is continuously used during state update, because the IV is public and the attacker will be able to evaluate the function $(\textit{volatile inner state}, IV) \longrightarrow \textit{keystream block}$ for randomly chosen volatile inner states and the proper IV. The only advantage of continuously using the IV would be that TMD tradeoff precomputations could be prevented as, if it is not a chosen-IV attack scenario, the attacker would have to wait for which IV he actually obtains the required/attacked keystream. ([HKMZ17])

In this note, we will argue why we believe that this restriction of continuous IV use to the context of Sprout-like ciphers was probably premature. More precisely, we will explain that, under certain conditions, general stream cipher constructions can benefit from this new design idea as well.

2 Continuously Using the IV with Stream Ciphers working in Packet Mode

At FSE 2017, the small-state stream cipher Lizard [HKM17] was introduced. Apart from its provable security against TMD-TO-based key recovery attacks, one of its prominent characteristics is that it explicitly targets *packet mode* scenarios, i.e., application contexts where only a moderate number of keystream bits needs to be generated per key/IV pair. More precisely, Lizard allows to generate up to 2^{18} keystream bits per key/IV pair, which would be sufficient for many existing communication scenarios like Bluetooth, WLAN or HTTPS as explained in [HKM17].

The authors of Lizard claimed 80-bit security against key recovery and 60-bit security against distinguishing. The lower security level against distinguishing results from the fact that, as pointed out by the designers in [HKM17], TMD-TO-based inner state recovery attacks like those of Babbage [Bab95] or Biryukov and Shamir [BS00] are still possible for Lizard, because after the state initialization of the cipher has been completed, the keystream is then generated solely in dependence of the resulting (121-bit wide) initial state, exactly as it is done by classical stream ciphers like Trivium [CP05] and Grain [HJMM08].

In the following, we will explain that by using a stream cipher in packet mode together with continuously involving the IV in the state update, TMD-TO-based inner state recovery attacks can actually be thwarted. For a technical discussion w.r.t. why the assumption of continuous IV availability for encryption purposes is actually plausible in many cases, we refer the reader to [HKMZ17].

2.1 Classical TMD-TO Attacks

Let us briefly revisit the relevant details of the well-known TMD-TO attacks of Babbage [Bab95] and Biryukov and Shamir [BS00]. We start with some terminology.

Definition 2.1

We consider the following complexities w.r.t. TMD-TO attacks against stream ciphers:

P: the preprocessing time of the attack,

T: the online time of the attack,

M: the memory required for the attack,

D: the (keystream) data required for the attack.

When we speak of the *overall complexity* of a TMD-TO attack, we refer to the maximum of the above four cost factors (including preprocessing time). Correspondingly, we assume attackers whose goal is to keep the overall complexity of their attack as low as possible.[a]

[a] Note that, when ignoring precomputation complexity, e.g., the eSTREAM portfolio [ECR08] member Grain v1 [HJM06] would have to be considered broken based on papers like [Bjø08].

Let us denote by N the number of all possible inner states of the targeted KSG. The TMD-TO attack of Babbage [Bab95] has the tradeoff curve $TM = N$ with $P = M$ and $T \leq D$ (where $T < D$ means that some of the available data is ignored during the online phase of the attack). Obviously, the best overall complexity that can be achieved here is $N^{1/2}$ by choosing $T = M = N^{1/2}$. The TMD-TO attack of Biryukov and Shamir [BS00] is based on Hellman's time-memory tradeoff attack for block ciphers [Hel80] and has the tradeoff curve $TM^2D^2 = N^2$ with $P = N/D$ and $T \geq D^2$. As, in our model, the overall complexity includes preprocessing, $P = N/D$ implies here that $N^{1/2}$ is a lower bound for the attack's overall complexity. This lower bound, however, cannot be achieved as $P = D = N^{1/2}$ would imply $T \geq N$ due to the restriction $T \geq D^2$. Note that, even if we did not consider preprocessing to be part of the overall attack complexity, the original attack of Biryukov and Shamir would not have an overall complexity lower than $N^{1/2}$ due to the condition $T \geq D^2$. Because choosing $T < N^{1/2}$ would imply $D < N^{1/4}$ and, in order to satisfy $TM^2D^2 = N^2$, also $M > N^{1/2}$.

In [BS00], Biryukov and Shamir also discuss a technique called *BSW-sampling*, which was used by Biryukov, Shamir, and Wagner in [BSW01] to attack the GSM cipher A5/1. While BSW-sampling allows to relax the restriction $T \geq D^2$ in the above attack, the tradeoff curve $TM^2D^2 = N^2$ and the relation $P = N/D$ remain unchanged. Hence, if one considers precomputation to be part of the overall attack complexity (as we do), even the use of BSW-sampling does not allow for attacks with overall complexity lower than $N^{1/2}$.

2.2 Attacking a Packet-Mode, Continuous-IV-Use Stream Cipher via Classical TMD-TO Attacks

Let us now consider an arbitrary KSG-based stream cipher with a volatile inner state size of 100 bits, a key size of 80 bits, and an IV size of 80 bits. Moreover, let us assume that this stream cipher is used in packet mode with a limit of 2^{20} keystream bits per key/IV pair and that, during keystream generation, the IV is continuously employed in the state update (like the key is continuously employed in the state update with Sprout-like ciphers). Then, a generic TMD-TO attacker in the spirit of [Bab95] and [BS00] has the following two possibilities:

1. He uses the fact that the IV is public and tries to invert the resulting function $F_{IV}(\textit{volatile inner state}) \longrightarrow \textit{keystream block}$ by randomly choosing volatile inner states of 100 bits and looking for a collision in the keystream of the attacked packet. As the data there is limited to 2^{20} bits, both Babbage and Biryukov-Shamir TMD-TO attacks will lead to overall complexities of at least 2^{80} (the cost of exhaustive key search).[1] In particular, observe that using data from another packet (generated under another initialization vector IV') leads to an independent birthday experiment, as the attacker then has to evaluate the different function $F_{IV'}$. That is, data obtained on the basis of different IVs cannot be used here to achieve an 'overall birthday bound-based' advantage.

2. He ignores the information about the IV and uses data from various packets, i.e., he tries to invert the function $F(\textit{volatile inner state}, IV) \longrightarrow \textit{keystream block}$. Then, however, he has to sample randomly from the space $\textit{Volatile Inner States} \times \textit{IV Space}$ of size 2^{100+80}, again leading to an overall attack complexity above that of exhaustive key search.

[1] More precisely, for the Babbage attack, the restrictions $T \leq D$ and $P = M$ together with the tradeoff curve $TM = N$ imply $M \geq 2^{80}$ and, hence, also $P \geq 2^{80}$ in our example. Similarly, for the Biryukov-Shamir attack, if $D \leq 2^{20}$, with or without BSW-sampling, due to $P = N/D$ the precomputation complexity is at least 2^{80} for $N = 2^{100}$.

Also note that the less common approach of attacking the function $F(Key, IV) \longrightarrow$ *keystream prefix* for a collection of keystream prefixes obtained on the basis of different IVs (as taken in, e.g., [HS05] and [DK08]), would not work, as well, because the key size and the IV size add up to 160 here, leading to an overall complexity of at least 2^{80}.[2]

We would like to point out that for scenarios where different (e.g., session) keys are used, it is important to deprive an attacker of the possibility to collect more data based on a situation where the same IVs are used in different sessions, as this would eventually allow him to recover a secret inner state (and, possibly, by clocking the cipher back, even the session key) of one of these sessions. Potential countermeasures are here to keep the IV counter over sessions (instead of resetting it) to choose the IVs at random. If IV reset is inevitable, one could also increase the volatile inner state from 100 bits to 120 bits in our above example and limit the number of session keys to a (in our opinion plausible) number of 2^{20}.[3] This would allow an attacker to obtain at most 2^{40} keystream bits generated under the same IV, and, for N now being 2^{120}, again lead to overall complexities of at least 2^{80} for the above classical TMD-TO attacks.

2.3 Towards Practical Instantiations of our Design Idea

While the target of this note is mainly to stimulate a general discussion about our new design idea, we still would like to point out an important aspect which has to be treated with special care in potential future instantiations: the concrete way in which the IV is involved in the state update.

As we have seen in 2.2, the security of our approach against generic TMD-TO attacks is based on the assumption that an attacker will not be able to use the same function F for attacking different packets. However, certain correlations between the functions F_{IV} and $F_{IV'}$ for different IVs IV and IV' are unavoidable. For example, given some arbitrary volatile inner state S, the keystream blocks $F_{IV}(S)$ and $F_{IV'}(S)$ will have the same first (i.e., leftmost) bit, as in our new design approach, the IV is continuously involved in the state update but does not enter the cipher's output function directly. Note that the same applies to Sprout-like ciphers w.r.t. the continuous use of the secret key.

In this note, we do not treat the potential security implications of such 'subtle' correlations, as we are focussing on generic TMD-TO attacks, which treat the function F as a black-box. In particular, we also do not consider cipher-specific approaches which might perform on-the-fly adaptions of F in order to make it applicable also to data obtained from other packets. However, in concrete instantiations, it might be required to fend off such non-generic attacks, which have yet to be developed.

But one important rule for future instantiations of our new design approach can be formulated right away: it is of vital importance that each bit of an IV IV must always have the potential to influence the keystream block $F_{IV}(S)$ generated on the basis of IV and a volatile inner state S. To illustrate this rule, imagine the bad *round IV function* $IV_{\text{Round}} = IV_{(2t) \bmod 80} \cdot IV_{(2t+1) \bmod 80}$ for an IV IV of length 80 bits and t denoting the corresponding keystream generation step of the KSG. In this case, a (chosen-IV) attacker could focus on the set (of size about $2 \cdot 2^{40}$) of all IVs which either have only zeros at their even index positions or have only zeros at their odd index positions. Given the above bad round IV function, all these IVs would never influence the state update during keystream generation, allowing an attacker to target all the corresponding keystream packets with a

[2] To avoid any misconceptions, we would like to point out that also Grain v1 [HJM06], which has key size 80 bits but IV size only 64 bits, would not succumb to this attack in a *single-key scenario*. This is due to the fact that under a fixed key, an attacker would only be able to collect at most 2^{64} keystream prefixes, leading to an overall complexity above 2^{80} for this type of TMD-TO attack. By choosing key length = IV length = 80 bits in our example, however, we can resist this attack even for *multiple-key scenarios*.

[3] Remember that when talking about state sizes of 100 or 120 bits, we are in the area of lightweight cryptography with its corresponding applications.

single function F (using the classical, generic TMD-TO attacks), thereby defeating our assumption about the limit of data available to him.

Also less extreme scenarios could be imagined, where an attacker knows that for certain blocks of a number of keystream packets, not all IV bits are involved in the underlying state updates. Note that, due to their 'complicated' *round key functions*, the continuous-key-use ciphers Sprout and Fruit have succumbed to similar TMD-TO attacks in the past w.r.t. the involvement of the key bits (see, e.g., [EK16] and [HKMZ17]). In fact, in [HKMZ17] it was hence conjectured that the round key function of Plantlet, which, in each step, simply cyclically XORs one key bit to the feedback bit of the NFSR, is actually the optimal one for thwarting generic TMD-TO attacks. In the same spirit, we also believe that, for instantiations of our new design approach, cyclically XORing one IV bit per step to the volatile inner state should be the way to go.

3 Conclusion

In this note, we presented a new idea for designing stream ciphers which resist TMD-TO inner state recovery attacks. It combines the concept of explicitly targeting packet mode scenarios (as introduced in [HKM17]) and the idea of continuously using the IV as part of the state update (as introduced in [HKMZ17]). While in [HKMZ17], continuously using the IV was suggested to protect Sprout-like ciphers (i.e., ciphers which continuously use the secret key) against TMD-TO *distinguishing* attacks, we have explained here that for stream ciphers used in packet mode, it can also thwart TMD-TO *inner state recovery* attacks. In particular, our approach alleviates the need for additional countermeasures like the continuous key use of Sprout-like ciphers or the second key addition in Lizard's state initialization algorithm. Let us point out, however, that this by no means reduces the significance of these other design paradigms. With this note, we simply seek to contribute an alternative approach for application scenarios where keystream packets of moderate size are required and the IV is continuously available for encryption purposes. As important further steps in this direction, we see the development of a concrete instantiation of our new approach as well as actually *proving* its security, e.g., similar to how the security of Lizard against TMD-TO key recovery attacks was proved in [HK15] or to how the security of Trivium and Grain against TMD-TO inner state recovery attacks was proved in [Kra17].

References

[AM15] Frederik Armknecht and Vasily Mikhalev. On Lightweight Stream Ciphers with Shorter Internal States. In Gregor Leander, editor, *Fast Software Encryption: 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, pages 451–470. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[Bab95] S.H. Babbage. Improved "exhaustive search" attacks on stream ciphers. In *Security and Detection, 1995., European Convention on*, pages 161–166, May 1995.

[Ban15] Subhadeep Banik. Some Results on Sprout. In Alex Biryukov and Vipul Goyal, editors, *Progress in Cryptology – INDOCRYPT 2015: 16th International Conference on Cryptology in India, Bangalore, India, December 6-9, 2015, Proceedings*, pages 124–139. Springer International Publishing, Cham, 2015.

[BB06] Elad Barkan and Eli Biham. Conditional Estimators: An Effective Attack on A5/1. In Bart Preneel and Stafford Tavares, editors, *Selected Areas in Crypto-*

graphy: 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers, pages 1–19. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[BD06] Steve Babbage and Matthew Dodd. The stream cipher MICKEY 2.0. eSTREAM: the ECRYPT Stream Cipher Project, 2006. http://www.ecrypt.eu.org/stream/p3ciphers/mickey/mickey_p3.pdf.

[BGW99] Marc Briceno, Ian Goldberg, and David Wagner. A pedagogical implementation of A5/1, 1999. Available at http://www.scard.org/gsm/a51.html.

[Bjø08] Tor E. Bjørstad. Cryptanalysis of Grain using Time/Memory/Date Tradeoffs. eSTREAM, ECRYPT Stream Cipher Project, Report 2008/012, 2008. http://www.ecrypt.eu.org/stream/papersdir/2008/012.pdf.

[BS00] Alex Biryukov and Adi Shamir. Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers. In Tatsuaki Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000: 6th International Conference on the Theory and Application of Cryptology and Information Security Kyoto, Japan, December 3–7, 2000 Proceedings*, pages 1–13. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[BSW01] Alex Biryukov, Adi Shamir, and David Wagner. Real Time Cryptanalysis of A5/1 on a PC. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, and Bruce Schneier, editors, *Fast Software Encryption: 7th International Workshop, FSE 2000 New York, NY, USA, April 10–12, 2000 Proceedings*, pages 1–18. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[CP05] Christophe De Cannière and Bart Preneel. Trivium – Specifications. eSTREAM: the ECRYPT Stream Cipher Project, 2005. http://www.ecrypt.eu.org/stream/p3ciphers/trivium/trivium_p3.pdf.

[DK08] Orr Dunkelman and Nathan Keller. Treatment of the Initial Value in Time-Memory-Data Tradeoff Attacks on Stream Ciphers. Cryptology ePrint Archive, Report 2008/311, 2008. http://eprint.iacr.org/2008/311.

[DR08] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008. Updated by RFCs 5746, 5878, 6176, 7465, 7507, 7568, 7627, 7685, 7905, 7919.

[ECR08] ECRYPT – European Network of Excellence for Cryptology. eSTREAM: the ECRYPT stream cipher project, 2008. http://www.ecrypt.eu.org/stream/.

[EK16] Muhammed F. Esgin and Orhun Kara. Practical Cryptanalysis of Full Sprout with TMD Tradeoff Attacks. In Orr Dunkelman and Liam Keliher, editors, *Selected Areas in Cryptography - SAC 2015: 22nd International Conference, Sackville, NB, Canada, August 12-14, 2015, Revised Selected Papers*, pages 67–85. Springer International Publishing, Cham, 2016.

[FMS01] Scott Fluhrer, Itsik Mantin, and Adi Shamir. Weaknesses in the Key Scheduling Algorithm of RC4. In Serge Vaudenay and Amr M. Youssef, editors, *Selected Areas in Cryptography: 8th Annual International Workshop, SAC 2001 Toronto, Ontario, Canada, August 16–17, 2001 Revised Papers*, pages 1–24. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[GHX16] Vahid Amin Ghafari, Honggang Hu, and Chengxin Xie. Fruit: Ultra-Lightweight Stream Cipher with Shorter Internal State. Cryptology ePrint Archive, Report 2016/355, 2016. http://eprint.iacr.org/2016/355.

[Hel80] M. Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, 26(4):401–406, Jul 1980.

[HJM06] Martin Hell, Thomas Johansson, and Willi Meier. Grain - A Stream Cipher for Constrained Environments. eSTREAM: the ECRYPT Stream Cipher Project, 2006. http://www.ecrypt.eu.org/stream/p3ciphers/grain/Grain_p3.pdf.

[HJMM08] Martin Hell, Thomas Johansson, Alexander Maximov, and Willi Meier. The Grain Family of Stream Ciphers. In Matthew Robshaw and Olivier Billet, editors, *New Stream Cipher Designs: The eSTREAM Finalists*, pages 179–190. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[HK15] Matthias Hamann and Matthias Krause. On Stream Ciphers with Provable Beyond-the-Birthday-Bound Security against Time-Memory-Data Tradeoff Attacks. Cryptology ePrint Archive, Report 2015/636, 2015. http://eprint.iacr.org/2015/636.

[HKM17] Matthias Hamann, Matthias Krause, and Willi Meier. LIZARD – A Lightweight Stream Cipher for Power-constrained Devices. *IACR Transactions on Symmetric Cryptology*, 2017(1):45–79, 2017.

[HKMZ17] Matthias Hamann, Matthias Krause, Willi Meier, and Bin Zhang. Design and analysis of small-state grain-like stream ciphers. *Cryptography and Communications*, Nov 2017.

[HS05] Jin Hong and Palash Sarkar. New Applications of Time Memory Data Tradeoffs. In Bimal Roy, editor, *Advances in Cryptology - ASIACRYPT 2005: 11th International Conference on the Theory and Application of Cryptology and Information Security, Chennai, India, December 4-8, 2005. Proceedings*, pages 353–372. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[Ins97] Institute of Electrical and Electronics Engineers. IEEE Standard for Information Technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-1997*, pages i–445, 1997.

[Ins04] Institute of Electrical and Electronics Engineers. IEEE Standard for Information Technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements. *IEEE Std 802.11i-2004*, pages 1–190, July 2004.

[Kra17] Matthias Krause. On the Hardness of Trivium and Grain with respect to Generic Time-Memory-Data Tradeoff Attacks. Cryptology ePrint Archive, Report 2017/289, 2017. https://eprint.iacr.org/2017/289.

[LMV05] Yi Lu, Willi Meier, and Serge Vaudenay. The Conditional Correlation Attack: A Practical Attack on Bluetooth Encryption. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005. Proceedings*, pages 97–117. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[LNP15] Virginie Lallemand and María Naya-Plasencia. Cryptanalysis of Full Sprout. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA,* *USA, August 16-20, 2015, Proceedings, Part I*, pages 663–682. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[MAM17] Vasily Mikhalev, Frederik Armknecht, and Christian Müller. On Ciphers that Continuously Access the Non-Volatile Key. *IACR Transactions on Symmetric Cryptology*, 2016(2):52–79, 2017.

[Pop15] A. Popov. Prohibiting RC4 Cipher Suites. RFC 7465 (Proposed Standard), February 2015.

[Sch95] Bruce Schneier. *Applied Cryptography (2nd Ed.): Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., New York, NY, USA, 1995.

[SIG14] Bluetooth SIG. Bluetooth Core Specification 4.2, 2014. https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=286439.

[ZG15] Bin Zhang and Xinxin Gong. Another Tradeoff Attack on Sprout-Like Stream Ciphers. In Tetsu Iwata and Hee Jung Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015: 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 – December 3, 2015, Proceedings, Part II*, pages 561–585. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.