State Convergence and the effectiveness of Time-Memory-Data Tradeoffs

Sui-Guan Teo, Kenneth Koon-Ho Wong, Ed Dawson
Information Security Institute
Queensland University of Technology
Brisbane Qld 4001, Australia
{sg.teo, kk.wong, e.dawson}@qut.edu.au

Leonie Simpson
Faculty of Science and Technology
Queensland University of Technology
GPO Box 2434, Brisbane Qld 4001, Australia
lr.simpson@qut.edu.au

Abstract—Various time-memory tradeoffs attacks for stream ciphers have been proposed over the years. However, the claimed success of these attacks assumes the initialisation process of the stream cipher is one-to-one. Some stream cipher proposals do not have a one-to-one initialisation process. In this paper, we examine the impact of this on the success of time-memory-data tradeoff attacks. Under the circumstances, some attacks are more successful than previously claimed while others are less. The conditions for both cases are established.

Index Terms—Stream cipher, Time-Memory-Data Tradeoffs, state convergence, A5/1, Mixer, ZUC

I. INTRODUCTION

Modern stream cipher applications use a secret key and a publicly known initialisation vector (IV) to form an initial internal state before keystream generation begins. This is commonly used in secure communications, where a single communication in frame-based applications can consist of multiple frames. A communication will use a single master key and each frame in the communication will be encrypted using that key and a distinct IV. For example, a mobile phone conversation is divided into many frames. Each frame in the communication is encrypted separately using the same master key and using the frame number as the IV. The initial state for each frame formed from the master key and IV is referred to as a session key. Given a suitable state size (at least equal to the sum of the key and IV lengths), a good initialisation process should ensure that each key-IV pair generates a session key and hence a distinct keystream.

State convergence in a keystream generator occurs when two distinct internal states generate the same next state. State convergence can occur either during initialisation, during keystream generation, or both. For the stream ciphers A5/1 [8], Mixer [9] and ZUC [10], analysis reveals that the keystream generators of each of these stream ciphers experience state convergence [11]–[14]. This occurs only during initialisation for Mixer but during both initialisation and keystream generation for A5/1 [13]. Where state convergence occurs during initialisation, the same keystream will be produced from different key-IV pairs.

A generic technique in stream cipher cryptanalysis is the time-memory-data tradeoff (TMDT) attack. TMDT attacks were first used to attack block ciphers by Hellman [1] and later adapted for stream cipher cryptanalysis by Babbage [2] and Golić [3]. Additional TMDT attacks on stream ciphers are proposed by Biryukov and Shamir [4], Hong and Sarkar [5], [6], and Dunkelman and Keller [7]. In this paper, we discuss the effectiveness of TMDT attacks when applied to keystream generators for which state convergence occurs, particularly with respect to the type of state convergence that occurs.

This paper is organised as follows. Section II gives a brief introduction to the initialisation and keystream generation process of keystream generators and reviews common TMDT on stream ciphers. Section III discusses the effect state convergence has on TMDT master and session key recovery attacks. Section IV concludes this paper and proposes possible areas for future research.

II. BACKGROUND

A. Initialisation and keystream generation process

Keystream generators for stream ciphers operate by maintaining an internal state and applying update and output functions to the state. The state of a keystream generator is of size $s$ bits. Modern keystream generators take two inputs: a master key and an IV, of size $k$ and $v$ bits, respectively. Thus, a key-IV pair has a total length of $k + v$ bits. Before keystream generation commences, a key-IV pair is used to form an initial internal state value. This process is referred to as initialisation and can be considered as a mapping from binary vectors of length $k + v$ to those of length $s$.

The initialisation process is often performed in three phases: key-loading, IV-loading and the diffusion phase. In the key-loading and IV-loading phases, the master key and IV are transferred to the keystream generator’s state. When both the master key and IV have been transferred, the stream cipher is in its “loaded state”. If $s < k + v$, key and IV loading results in state compression and consequently, the total number of distinct keystreams is less than the total number of key-IV pairs. If $s \geq k + v$, the loading process potentially provides $2^{k+v}$ distinct loaded states.

Following this, the diffusion phase begins. This is generally the most complex phase of the initialisation process and it is important, as using the loaded state directly to begin keystream generation could make the stream cipher vulnerable to correlation or algebraic attacks. The diffusion phase consists of a
number of iterations of the initialisation state-update function. Each iteration of the initialisation state-update function can be considered as a function which maps the state space to itself. This mapping should be one-to-one and nonlinear in nature. After the initialisation process is complete, the keystream generator is said to be in an initial state. Let \( I \) be the total number of distinct initial states. If \( s \geq k + v \) and the initialisation process is well-designed, \( I = 2^{k+v} \). If \( I < 2^{k+v} \), state convergence has occurred during initialisation.

Once the keystream generator is in its initial state, the keystream generation phase begins. To generate keystream, a state-update function is applied to the internal state of the stream cipher and an output function is applied to this internal state. This state-update function can be either be the same function used in the initialisation phase or a different function. An example of a stream cipher which uses the same state-update function for both initialisation and keystream generation is A5/1 [8], while Mixer [9] is an example of a stream cipher which uses a different state-update function for initialisation and keystream generation.

To prevent TMDT attacks, it is recommended that modern stream ciphers have an internal state size which is greater or equal to \( k + v \), where \( v \geq 1.5k \) [7]. Since the state space is at least the size of the space spanned by a key-IV pair, it is reasonable to expect that the initialisation process will be one-to-one, that is, each distinct key-IV pair should map to a distinct state at the end of initialisation.

**B. Review of some time-memory-data tradeoff attacks**

The goal of TMDT attacks is to recover either the session key, the internal state at a known point in time or the master key. If the attacker manages to recover the session key of a keystream generator, they can use it to generate keystream to decrypt the entire frame. However, the attacker will not be able to use this session key to decrypt other frames in the conversation, since these will have been encrypted using the same master key but a different IV. Master key recovery is stronger as this allows an attacker to decrypt all other frames in a conversation.

TMDT attacks are performed in two phases: the pre-computation phase and the online phase. In the pre-computation phase, a lookup table is constructed. This table has two columns. For state recovery, the first column consists of selected session keys (initial internal states) of the stream cipher. For master key recovery, the first column consists of selected master keys (and might also include the IV). In both scenarios, the second column consists of a segment of keystream generated using either the corresponding key-IV pair, session key or internal state. In the online-phase of the attack, the attacker compares the captured keystream to the second column of the lookup table. If a match is detected, the attacker assumes that the obtained session key, internal state or key-IV pair is correct.

The complexity of a TMDT attack can be described using a series of variables. \( D \) is the amount of data the attacker needs in the online-phase of the attack to recover the master key. \( P \) is the pre-computation time needed to construct the lookup table. \( M \) is the memory needed to construct and store the table. During the online phase, the attacker attempts to recover the session or master key by searching through the lookup table. The time taken to do the search is denoted by \( T \). The success of the attack depends on \( T \) or \( M \) or the sum of \( T + M \) being less than \( 2^k \) or \( 2^v \), depending on the particular TMDT attack being used. Since \( P \) is a one-off operation, it is assumed that the attacker has already pre-computed the lookup table beforehand and the time taken for this operation is not considered when measuring the complexity of the TMDT attack. In this section, we review the major TMDT attacks on stream ciphers.

1) **Babbage and Golić.** Babbage [2] and Golić [3] independently applied the TMDT attacks to stream ciphers. Their session key recovery attack is referred to as the BG attack in the remainder of this paper.

In the pre-computation phase, an attacker selects either \( M \) different session keys or internal states. For each of these, the attacker produces some keystream of length \( s \). The attacker then stores the session key-keystream pair in a lookup table, sorted according to the keystream.

In the real-time phase of the attack, the attacker takes a segment of keystream of length \( D + \log s - 1 \) they have captured and uses a sliding-window to produce all \( D \) possible keystream sub-strings of length \( s \). The attacker then searches the lookup table to see if any of these substrings match. If there is a match, the session key corresponding to the keystream sub-string is considered to be the session key which generated the captured keystream. If the TMDT satisfies the following equations

\[
T \cdot M = 2^P \quad \text{with} \quad P = M
\]

the attack complexity is less than that of exhaustive keysearch. To provide resistance to this attack, both Babbage and Golić recommend that the size of the internal state of the stream cipher should be at least twice the key size.

2) **Biryukov and Shamir.** Biryukov and Shamir [4] combine the concepts of Hellman’s TMDT attack on block ciphers [1] and the BG attack to provide a more efficient TMDT attack on stream ciphers. Their session key recovery attack is referred to as the BS attack for the remainder of this paper.

The pre-computation phase of the BS attack is similar to Hellman’s pre-computation phase. The attacker defines a function \( f \), which generates the keystream in the stream cipher. The attacker also chooses random permutations to take place of the function \( h \). \( h \) is a function which maps the \( s \)-bit state to another \( s \)-bit state. The attacker defines \( g = h \circ f \) and creates lookup tables using Hellman’s lookup table construction method. In the online phase, the attacker uses any instance \( c \) in the \( D \) keystreams obtained and iteratively applies \( g \) to \( h(c) \) until the \( s \)-bit value \( h(c) \) matches a entry in the second column of the lookup table. Once a match is found, the session key which generated the keystream is recovered using the method used in Hellman’s online phase attack. The tradeoff curve of the
BS attack is given by:
\[ T \cdot M^2 \cdot D^2 = 2^{2s} \quad \text{with} \quad P = \frac{2^2s}{D} \quad \text{and} \quad 1 \leq D^2 \leq T \quad (2) \]

The BS attack reiterates the importance that the size of the internal state of a keystream generator needed to be at least twice the master key size so that TMD tradeoffs are worse than exhaustive keysearch.

3) Hong and Sarkar: Hong and Sarkar’s TMDT attack [5], [6] aims to recover the master key, as opposed to recovering the internal state or session key in the earlier attacks. Their master key recovery attack will be referred to as the HS attack for the remainder of this paper.

In the pre-computation phase of the HS attack, the attacker first chooses random key-IV pairs, storing the master key and the IV in the first column of the lookup table. For each key-IV pair, the attacker generates a keystream of length \( k + v \) bits. The tradeoff curve from the HS attack is the same as BS curve, but instead of it being an internal-state to keystream mapping, the HS attack uses a key-IV to keystream mapping:
\[ T = M = 2^{2s(k+v)} \quad \text{with} \quad D = 2^{2s(k+v)} \quad (3) \]

Thus, if the attacker has access to \( D = 2^{\frac{1}{2}(k+v)} \) bits of keystream, the attacker can recover the master key with a time and memory complexity of \( T = M = 2^{\frac{1}{2}(k+v)} \). If \( v < k \), the complexity of the attack is less than exhaustive key search.

In order to resist the HS attack, Hong and Sarkar recommend that the IV size is at least as long as that of the master key.

4) Dunkelman and Keller: The TMDT attack by Dunkelman and Keller [7], referred hereafter as the DK attack is a master key recovery attack. In the DK attack, an attacker constructs lookup tables for chosen IVs. This approach is different from the HS attack, where each lookup table would consist of arbitrary IVs. Constructing lookup tables for each IV allows the attacker to take advantage of the fact the IV is a publicly known value.

By constructing tables for specific IVs, the following tradeoff curve is obtained.
\[ T \cdot M^2 \cdot D^2 = 2^{2(k+v)} \quad (4) \]

Note that this is the same tradeoff curve as the HS and BS attack. However, because of the IV table-based approach, this approach does not use multiple keystreams and hence, imposes no restrictions on the parameters. Therefore, even for \( T = M = D \), the complexity of the attack is less than exhaustive key search as long as \( 2^{2v} < 2^{3k} \). That is, a stream cipher would be resistant to the DK attack if the \( v \geq 1.5k \).

III. STATE CONVERGENCE AND THE EFFECTIVENESS ON TMD TRADEOFF ATTACKS

Recent analysis of several stream ciphers, namely A5/1 [8], Mixer [9] and ZUC [10], revealed that the initialisation processes are not one-to-one. For A5/1 and Mixer, the choice of state-update functions means that the number of distinct initial states decreases as the number of iterations of the state-update function increases [3], [11], [13]. There are three possible scenarios for state convergence in keystream generators. They are:

- **Scenario 1.** The same master key used with different IVs generates the same initial state.
- **Scenario 2.** The same IV used with different master keys generates the same initial state.
- **Scenario 3.** Distinct key-IV pairs generate the same initial state.

We now analyse the effect state convergence has on the success of session key recovery and master key recovery TMDT attacks. A summary of our findings can be found in Table I. Table entries are either a \( \checkmark \) or \( ? \), where a \( \checkmark \) means an attacker can be confident that the session key or master key they recovered is correct, while a \( ? \) means there is the possibility that the attacker has not recovered the correct master key.

A. Effect on TMDT attacks on stream ciphers.

1) Session key recovery: The lookup tables for session key TMDT attacks are constructed so that an attacker can recover the session key for a particular keystream. If the captured keystream can be found in the lookup table, the attacker can use the corresponding session key to generate sufficient keystream to decrypt the entire encrypted frame.

An alternative to session key recovery is the internal state recovery TMDT attack. An internal state TMDT attack recovers the internal state of a keystream generator at a known point in time during keystream generation. The process used in internal state recovery TMDT attacks is similar to session key recovery. Note that although the process is the same, there is the possibility that internal state recovery attacks can only recover a portion of the frame, compared to session key recovery’s ability to decrypt the entire frame. We only consider session key recovery in this section, although in some cases it might actually be the internal state that the attacker is recovering.

State convergence has a positive impact on the success of attacks aimed at session key recovery. If an attacker recovers the session key, they will be able to correctly decrypt the entire frame. It does not matter which key-IV pair generated the session key, since the definition of state convergence, multiple key-IV pairs can generate the same session key. Hence, the three different scenarios described have the same outcome with respect to the success or failure of TMDT attacks which recover session keys. However, since the attacker does not know the master key that was used, they will not be able to decrypt other frames in the communication and will need to perform the online phase of the TMDT attack again in order to decrypt these. If the number of distinct session keys \( I \) is such that \( I < 2^{k+v} \), the tradeoff equation in Equation 2 will result in reduced time, memory and data requirements if the attacker is aware of this and constructs the lookup table accordingly.

We now present an example of how this reduced session key size may have a positive effect on session key recovery on an actual cipher which has the state convergence problem. Mixer is an example of stream cipher which uses different state-update functions for initialisation and keystream generation.
Mixer uses a 128-bit master key and a 64-bit IV to initialise a 217-bit internal state. Teo et al. [13] estimate that the total number of distinct session keys after all possible key-IV pairs undergoes the initialisation phase to be bounded by $2^{191}$ and $2^{108.99}$.

Applying Equation 2 without taking into account state convergence gives the following tradeoff: $T = M = 2^{96}$, and $D = 2^{48}$. This constitutes an attack on Mixer. The bounds when the total number of distinct session keys is $2^{191} - 2^{109}$ can be seen in Table II. In both cases, the time, memory, and data complexities may see significant reductions in time, memory, and data complexities.

2) Master Key Recovery and Scenario 1.: When a single master key and different IVs generate the same keystream, the HS attack can recover the correct master key if that key-IV pair was one of those selected for the construction of the lookup table. This is the case regardless of whether the state-update function used during initialisation and keystream generation is the same function or a different one. After the online phase of the TMDT attack, an attacker can check that the recovered IV is the same as that captured along with the keystream. If it is the same IV, the attacker can be confident that they have recovered the correct master key and can use that master key with other IVs to decrypt other frames in the communication. If the IV does not match the one recorded in the table, the attacker knows that they have recovered the wrong master key and would need to perform the online phase of the attack again.

A similar process happens in the DK attack. In the online phase, the attacker uses the appropriate IV-based lookup table and checks if there is a match on the captured keystream. If there is a match, the corresponding master key is the correct master key which generated the captured keystream. The attacker can then use that same master key with other IVs to decrypt other encrypted frames in the communication.

We now present an example of how this reduced master key size may have a positive effect on master key TMDT attacks on an actual cipher which has the state convergence problem. The ZUC stream cipher uses a 128-bit master key and a 128-bit IV. The keystream generator has a total state space of $s = 560$ bits. Since $k = v = 128$ bits and $k + v < \frac{s}{2}$, ZUC is resistant to most forms of TMDT attack except the DK attack, if state convergence is not taken into account. The tradeoffs for the HS and DK attack without taking into account the state convergence issue are $T = M = 2^{126}$ and $D = 2^{64}$; and $T = M = D = 2^{102.4}$ respectively. Wu et al. [14] note that ZUC had Scenario 1 state convergence. Consequently, the effective master key size is potentially reduced to 66 or 100 bits, depending on which differential attack was used. As Wu et al. made no mention of the effective IV size as a result of the state convergence, we assume that the effective IV size is still 128 bits. The new tradeoffs can be seen in Table III.

As can be seen from the new tradeoff equations, considering the state convergence, the ZUC stream cipher is now vulnerable to the HS attack. Furthermore, both the HS and DK attack now have significant reductions in time, memory and data requirements.

3) Master Key Recovery and Scenario 2.: Where the same keystream is generated by different master keys for any given IVs, the attacker does not have the confidence that master key they recovered is the correct one.

Let us assume that three master keys, $K_1$, $K_2$ and $K_3$, with the same IV, $V_1$, produce the same keystream and two master keys, $K_2$ and $K_3$ were used for the construction of the lookup table. The original frame was encrypted with the $K_1$:$V_1$ key-IV pair. During the online phase of the attack, the attacker recovers two keys $K_2$ and $K_3$, that with IV $V_1$ will produce the same keystream. If an attacker, incorrectly assuming that $K_2$ was the actual master key, tries to use $K_2$ to decrypt other frames, it should not be successful, since $K_2$ with a different IV $V_2$ will most likely not generate the same keystream as would have been generated by the $K_1$:$V_2$ pair. Since $K_1$ was not selected during the construction of the lookup table, the master key recovery TMDT attack in this scenario is equivalent to session key recovery. For the attack to succeed, the attacker has to hope that the correct master key was selected during the construction of the lookup table. If the correct master key was not used, the attacker will not be able to decrypt other encrypted frames in a single conversation.

<table>
<thead>
<tr>
<th>Key recovery</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session key recovery</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Master key recovery</td>
<td>HS: ✔</td>
<td>?</td>
<td>HS: ✔</td>
</tr>
</tbody>
</table>

TABLE I
SUMMARY TABLE ON THE EFFECTIVENESS OF TMDT ATTACKS.

<table>
<thead>
<tr>
<th>Original tradeoffs</th>
<th>New tradeoffs</th>
</tr>
</thead>
<tbody>
<tr>
<td>192 bits</td>
<td>191 bits</td>
</tr>
<tr>
<td>109 bits</td>
<td></td>
</tr>
</tbody>
</table>

TABLE II
ORIGINAL AND NEW TRADEOFFS FOR MIXER USING BIRYUKOV AND SHAMIR’S TRADEOFF EQUATION

<table>
<thead>
<tr>
<th>$T$</th>
<th>$M$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{96}$</td>
<td>$2^{96}$</td>
<td>$2^{48}$</td>
</tr>
<tr>
<td>$2^{96}$</td>
<td>$2^{96}$</td>
<td>$2^{48}$</td>
</tr>
<tr>
<td>$2^{96}$</td>
<td>$2^{96}$</td>
<td>$2^{48}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$T$</th>
<th>$M$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{96}$</td>
<td>$2^{96}$</td>
<td>$2^{48}$</td>
</tr>
<tr>
<td>$2^{96}$</td>
<td>$2^{96}$</td>
<td>$2^{48}$</td>
</tr>
<tr>
<td>$2^{96}$</td>
<td>$2^{96}$</td>
<td>$2^{48}$</td>
</tr>
</tbody>
</table>

The attacker can then use that same master key with other IVs to decrypt other encrypted frames in the communication.

We now present an example of how this reduced master key size may have a positive effect on master key TMDT attacks on an actual cipher which has the state convergence problem. The ZUC stream cipher uses a 128-bit master key and a 128-bit IV. The keystream generator has a total state space of $s = 560$ bits. Since $k = v = 128$ bits and $k + v < \frac{s}{2}$, ZUC is resistant to most forms of TMDT attack except the DK attack, if state convergence is not taken into account. The tradeoffs for the HS and DK attack without taking into account the state convergence issue are $T = M = 2^{126}$ and $D = 2^{64}$; and $T = M = D = 2^{102.4}$ respectively. Wu et al. [14] note that ZUC had Scenario 1 state convergence. Consequently, the effective master key size is potentially reduced to 66 or 100 bits, depending on which differential attack was used. As Wu et al. made no mention of the effective IV size as a result of the state convergence, we assume that the effective IV size is still 128 bits. The new tradeoffs can be seen in Table III.

As can be seen from the new tradeoff equations, considering the state convergence, the ZUC stream cipher is now vulnerable to the HS attack. Furthermore, both the HS and DK attack now have significant reductions in time, memory and data requirements.

3) Master Key Recovery and Scenario 2.: Where the same keystream is generated by different master keys for any given IVs, the attacker does not have the confidence that master key they recovered is the correct one.

Let us assume that three master keys, $K_1$, $K_2$ and $K_3$, with the same IV, $V_1$, produce the same keystream and two master keys, $K_2$ and $K_3$ were used for the construction of the lookup table. The original frame was encrypted with the $K_1$:$V_1$ key-IV pair. During the online phase of the attack, the attacker recovers the two keys $K_2$ and $K_3$, that with IV $V_1$ will produce the same keystream. If an attacker, incorrectly assuming that $K_2$ was the actual master key, tries to use $K_2$ to decrypt other frames, it should not be successful, since $K_2$ with a different IV $V_2$ will most likely not generate the same keystream as would have been generated by the $K_1$:$V_2$ pair. Since $K_1$ was not selected during the construction of the lookup table, the master key recovery TMDT attack in this scenario is equivalent to session key recovery. For the attack to succeed, the attacker has to hope that the correct master key was selected during the construction of the lookup table. If the correct master key was not used, the attacker will not be able to decrypt other encrypted frames in a single conversation.
If $\gamma$ is the number of master keys an attacker obtains at the end of the online phase of the attack and $\epsilon$, with $\epsilon < V$, being the effective IV size of the stream cipher, the HS and DK tradeoff curve would be

$$\left(T + \gamma\right) \cdot M^2 \cdot D^2 = 2^{(k + \epsilon)}$$

(5)

The memory and data requirements however, remain the same as would have been obtained in Equation 2. However, since $\gamma$ possible master keys can now appear in the lookup table, an attacker needs to try, on average, $\frac{2^\gamma}{\gamma}$ keys with other IVs before they can be certain if the master key they are currently trying is correct.

If a keystream generator uses the same state-update function for initialisation and keystream generation, it can be viewed as a keystream generator which performs an extended version of the initialisation process to generate keystream. Hence, if the segment of keystream used during the construction of the lookup table matched a segment of keystream which was capture not from the beginning of keystream generation, the number of possible master keys which could have generated the keystream with the particular IV can increase. Therefore, a successful TMDT master key recovery attack with keystream generator which use the same state-update function for both initialisation and keystream generation can be less likely than an attack on a keystream generator which uses a different state-update function for initialisation and keystream generation.

4) Master Key Recovery and Scenario 3.: Where the keystream is generated by different key-IV pairs, during the online phase an attacker will know if they have recovered the correct master key based on the publicly known IV. If $\tau$ and $\epsilon$, with $\tau < 2^k$ and $\epsilon < 2^v$, are the effective master key and IV size respectively, the tradeoff curve would be

$$T \cdot M^2 \cdot D^2 = 2^{(\tau + \epsilon)}$$

(6)

Similar to Scenario 1, since the attacker knows the IV used to generate the captured keystream and assuming the master key used to generate the captured keystream was used during the construction of the lookup table, the attacker can be confident that the master key they recover during the online phase of the TMDT attack is the correct one. Furthermore, since $\tau < 2^k$ and $\epsilon < 2^v$, the HS and DK attacks will be less than exhaustive master key search.

Biryukov et al.’s [12] TMDT attack on A5/1 describes an attack which is able to recover the master key in a few minutes at most. The most expensive cost of this attack is the precomputation complexity, which they calculated to be

$$P = M \cdot \sqrt{T} = 2^{48}$$

where $M = 2^{36}$, $T = 2^{24}$, and $2^{48}$ is the total number of initial states which will produce a certain 16-bit output prefix $(2^{64} \times 2^{-16})$. The estimates provided by Teo et al.’s [13] indicates the possibility that the number of distinct initial states at the end of A5/1’s diffusion phase to be $19.2/100 \times 2^{64} \approx 2^{61.62}$ due to state convergence. Using this new estimate, the precomputation complexity of Biryukov et al’s attack is reduced to $2^{61.62} \times 2^{-16} = 2^{45.62}$. This in turn, potentially reduces the time and memory requirements to be $M = 2^{35}$ and $T = 2^{21.24}$.

IV. DISCUSSION AND CONCLUSION.

This paper has analysed how state convergence could affect the effectiveness of TMDT attacks. In the case of session key TMDT attacks, an attacker potentially needs to guess a smaller set of session keys than what was originally intended by the designers of the stream cipher, since not all distinct key-IV pairs generate a distinct initial state. This could result in less time, data and memory requirements needed for the session key TMDT attacks to succeed than previously estimated by the designers of the stream cipher. The disadvantage of session key recovery TMDT attacks is if the attacker wanted to decrypt other encrypted frames in the communication, they would need to re-run the TMDT attack for each frame. If the attacker were to repeatedly use session key recovery TMDT attacks to decrypt multiple frames, it can be less efficient than master key recovery TMDT attacks.

For master key recovery TMDT attacks, the success of the attacks depend on the type of scenario, as outlined in Section III. Unless the convergence is such that different master keys with the same IV produce the same session key, an attacker who recovers a master key can check if the associated IV value matches which was observed along with the keystream. If the IV value is the same, the attacker can be confident that they have recovered the correct key. However, if state convergence was such that it was possible that multiple distinct master keys with the same IV generate the same keystream, there is a possibility that the master key recovered by the attacker during the online phase of the TMDT attack is not the correct master key. In this case, it is likely that

<table>
<thead>
<tr>
<th></th>
<th>HS</th>
<th>New HS</th>
<th>DK</th>
<th>New DK</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>128 bits</td>
<td>100 bits</td>
<td>66 bits</td>
<td>128 bits</td>
</tr>
<tr>
<td>$T$</td>
<td>$2^{128}$</td>
<td>$2^{114}$</td>
<td>$2^{97}$</td>
<td>$2^{102.4}$</td>
</tr>
<tr>
<td>$M$</td>
<td>$2^{128}$</td>
<td>$2^{114}$</td>
<td>$2^{97}$</td>
<td>$2^{102.4}$</td>
</tr>
<tr>
<td>$D$</td>
<td>$2^{64}$</td>
<td>$2^{57}$</td>
<td>$2^{48.5}$</td>
<td>$2^{102.4}$</td>
</tr>
</tbody>
</table>

TABLE III
ORIGINAL AND NEW TRADEOFFS FOR ZUC v1.4
the attacker can only decrypt a single frame and master key TMDT attacks may be less effective than claimed. The attacker can only be confident that they have actually recovered the correct master key if they can use it to decrypt the contents of all the frames in the communication.

In this paper, we included examples of ciphers which experience state convergence. The implementation of TMDT attacks on these ciphers to verify the expected reductions in time, memory and data requirements remains future work. Additionally, we plan to investigate other stream ciphers with potential state convergence problems.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful comments.

REFERENCES


