Al-Farouk journal of sciences
Volume (2), Issue (4) (Spt, 2026)
ISSN: 3135-2359

مجلة الفاروق للعلوم
المجلد (2) العدد 4 سبتمبر 2026
ردمد: 3135-2359

Comparative Analysis of 128-bit Key Loading Techniques for the A5/1 Stream Cipher: Statistical, Structural and Performance Evaluation

Sarah Farag A. Hmeid[1], and Fardous Mohamed A. Eljadi[2]
[12.] Department of Computer Science, Faculty of Science, University of Tripoli, Tripoli, Libya
Eng: Sarah Farag A. Hmeid: S.Hamid@uot.edu.ly
Received: 15/8/2026 Revised: 17/8/2026 Accepted: 05/9/2026 Publication: 25/10/2026

Abstract:

A5/1 is an encryption scheme that continues to be employed in the old-fashioned GSM network. It has a compact hardware architecture that makes it ideal for inexpensive modifications, but the size of its 64-bit key is not sufficient in view of modern computing capabilities. In this work, the performance of four methods of loading a 128-bit key in the same registers of A5/1 is examined: direct extension (D), extended warm-up with key salt (WS), key scrambling (SC) and state mixing (SX). The randomness of these variations was tested with the NIST SP 800-22 test suite applied to 100 sequences of 1000000 bits. The algorithm A5/1 is still used as the cryptographic scheme in the traditional GSM communication system. It has a small hardware structure, which makes it easy to implement any inexpensive alterations; however, the size of the 64-bit key is inadequate in the terms of contemporary computing resources. This paper evaluates four techniques for loading a 128-bit key into the registers of A5/1 without changing their structure: direct extension (D), extended warm-up with key salt (WS), key scrambling (SC), and state mixing (SX). Randomness was assessed for each technique with the NIST SP 800-22 test suite over 100 sequences of 10^6 bits, and every one of them passed. The techniques were then compared on two further measures: how fully they draw on the 128-bit key, and what initialization costs they incur. D and SC leave part of the key unused, since their effective key space does not exceed 2^{64}, whereas WS and SX exploit all 128 key bits. SX achieves this while retaining the original 186-cycle initialization and 93% of the original throughput; WS, by contrast, requires 10,086 cycles. SX is therefore the recommended technique. Because the internal state remains 64 bits, the proposal improves key loading at low cost, but it does not amount to a complete security solution.

Keywords: A5/1; stream cipher; key loading; equivalent keys; NIST SP 800-22; lightweight cryptography; GSM.

مقارنة تقنيات تحميل مفتاح بطول 128 بت في خوارزمية التشفير الانسيابي A5/1 من الجوانب الإحصائية والبنيوية والأدائية

سارة فرج حميد[1]، د. فردوس محمد الجدي[2]
[1.2] قسم الحاسب الآلي، كلية العلوم، جامعة طرابلس، طرابلس، ليبيا

الملخص:

تعدّ خوارزمية A5/1 نظامًا للتشفير لا يزال مستخدمًا في شبكات GSM التقليدية والقديمة. ويتميز ببنية عتادية بسيطة وصغيرة، مما يجعله مناسبًا لإجراء تعديلات منخفضة التكلفة .ومع ذلك، فإن طول مفتاحه البالغ 64بتًا لم يعد كافيًا في ظل قدرات الحوسبة الحديثة. تتناول هذه الدراسة تقييم أربع طرق لتحميل مفتاح بطول 128 بتًا في سجلات A5/1 نفسها دون تغيير بنيتها، وهي :التوسيع المباشر (D)، وفترة التهيئة الممتدة باستخدام ملح المفتاح (WS)، وتشويش المفتاح (SC)، ومزج الحالة (SX) . تم تقييم العشوائية في كل طريقة باستخدام مجموعة الاختبارات الإحصائية NIST SP 800-22، وذلك بتطبيقها على 100سلسلة، تحتوي كل منها على (10^6) بت، وقد اجتازت جميع الطرق الاختبارات الإحصائية. بعد ذلك، جرت مقارنة الطرق وفق معيارين إضافيين :مدى الاستفادة من كامل المفتاح ذي الـ128 بتًا، وتكلفة عملية التهيئة .أظهرت النتائج أن طريقتي التوسيع المباشر (D) وتشويش المفتاح (SC) لا تستفيدان من المفتاح بالكامل، إذ لا يتجاوز فضاء المفاتيح الفعلي لكل منهما (2^64)، في حين تستفيد طريقتا فترة التهيئة الممتدة (WS) ومزج الحالة (SX) من جميع بتات المفتاح البالغ عددها 128بتًا. وتحقق طريقة مزج الحالة (SX) هذه الاستفادة مع الاحتفاظ بدورة التهيئة الأصلية البالغة 186 دورة، والحفاظ على 93% من معدل الإنتاجية الأصلي .في المقابل، تتطلب طريقة فترة التهيئة الممتدة (WS)عددًا قدره 10,086دورة .وبناءً على ذلك، تُوصي الدراسة باستخدام طريقة مزج الحالة (SX). ونظرًا إلى أن الحالة الداخلية للخوارزمية لا تزال بطول 64 بتًا، فإن الطريقة المقترحة تحسّن آلية تحميل المفتاح بتكلفة منخفضة، لكنها لا تمثل حلًا أمنيًا متكاملاً.

الكلمات المفتاحية: A5/1؛ التشفير الانسيابي؛ تحميل المفتاح؛ المفاتيح المتكافئة؛ NIST test؛ التشفير خفيف الوزن؛ GSM.

1. Introduction

Stream ciphers are fast cryptographic primitives, encrypting data by combining it with a pseudorandom keystream. This makes them well suited to environments with restricted resources, such as mobile devices, IoT devices and embedded systems (Mammeri, 2024; Stallings, 2017). A5/1 is the stream cipher that was used to protect voice and data on the GSM air interface. It consists of three linear feedback shift registers (LFSRs), clocked irregularly under a majority rule (Briceno et al., 1999). Several attacks on the cipher have been published (Biham & Dunkelman, 2000; Biryukov et al., 2001; Golić, 1997; Mahalanobis & Shah, 2014), yet A5/1 remains in use. Passive measurements in 2025 confirmed that it is still negotiated on live 2G networks (Koch et al., 2025; Morgan,

2025), and research on reducing interception risk in GSM continues (Ahmad et al., 2025). The main shortcoming of A5/1 is the usage of a 64 bit session key, which can be attacked by brute-force attack. The solution is to make the key longer. This has been done in various studies by increasing the size of the registers (Fauzi et al., 2018) or adding more registers (Ahmad et al., 2025; Joshi & Mishra, 2021). The problem with these solutions is that it increases the cost of the hardware and that is a problem in the constrained environments where A5/1 algorithm designs are used (Li et al., 2023; Turan et al., 2025). In this paper we look at a lower cost alternative where the three registers with 19, 22 and 23 bits and the majority clocking are left as they are and only the method of loading the 128bit key is changed.

It should be noted that a longer key does not necessarily provide more security. If the loading procedure compresses 128 key bits into a 64-bit state, many keys will lead to the same state, and an attacker can search the state instead of the key. Previous key-extension proposals for A5/1 were evaluated mainly or only by statistical randomness tests (Fauzi et al., 2018; Jawad et al., 2024; Prajapat et al., 2022). This paper shows that such tests cannot detect this problem.

Contributions. The main contributions of this paper are as follows:

1. Four 128-bit key-loading techniques for unchanged A5/1 hardware (D, WS, SC and SX) are specified precisely as algorithms.
2. The four techniques are evaluated using the complete NIST SP 800-22 suite with both second-level criteria. It is shown that all of them pass and that P-values should not be used to rank them.
3. A structural analysis is provided to determine the effective key space of each technique, and it is confirmed experimentally by an explicit pair of equivalent keys and by ignored key bits.
4. The initialization cost and throughput are measured, and SX is identified as the only technique that uses the full key at almost the original cost.

The rest of this paper is organized as follows: Section 2 reviews the related work. Section 3 describes the A5/1 stream cipher. Section 4 introduces the four key-loading techniques. Section 5 outlines the experimental setup, and Section 6 presents the results. Section 7 discusses the results and the limitations of this study. The final section concludes this paper.

2. Related Work

The internal structure of A5/1 became public through reverse engineering (Briceno et al., 1999). Since then, several attacks have been proposed, such as guess-and-determine and time–memory trade-off attacks (Biham & Dunkelman, 2000; Biryukov et al., 2001; Golić, 1997), an improved guess-and-determine attack (Mahalanobis & Shah, 2014) and a side-channel attack (Jurecek et al., 2019). In addition, time–memory–data trade-off (TMDTO) analysis (Babbage, 1995; Biryukov & Shamir, 2000) provides a general design rule that applies directly to this work: the internal state of a stream cipher should be at least twice as long as its key. A5/1, with its 64-bit state, does not satisfy this rule even for its original key.

Several studies have been carried out to strengthen A5/1. Fauzi et al. (2018) made the registers larger to 40, 43 and 45 bits, which gives a 128-bit state, and evaluated the result using the full NIST suite. However, the feedback remained linear and the hardware cost increased. Joshi and Mishra (2021) used a 128-bit key with 23 LFSRs arranged and selecting batches in dynamic way. Their design produced good randomness, but at a large hardware cost and without standard statistical validation. Prajapat et al. (2022) modified the feedback functions to increase the randomness of the keystream and reported NIST results. Jawad et al. (2024) replaced the linear feedback with a 5-bit S-box, but only a reduced 16-bit model was analyzed with a small set of custom tests. Ahmad et al. (2025) increased the number of LFSRs to five, seven or nine and modeled the resulting reduction in interception probability.Two gaps can be observed in these studies. First, security is evaluated mainly through statistical randomness. Second, when the key is extended, it is not checked whether the extra key bits actually affect the keystream, that is, whether the effective key space increases. This paper addresses both gaps for the low-cost case in which the A5/1 registers are not changed.

3. The A5/1 Stream Cipher

A5/1 is made up of three LFSRs, R_1, R_2 and R_3, with lengths of 19, 22 and 23 bits (64 bits in total). Their feedback polynomials are defined in Eq. (1).

$$x^{19} + x^{18} + x^{17} + x^{14} + 1, \quad x^{22} + x^{21} + 1, \quad x^{23} + x^{22} + x^{21} + x^{8} + 1. \qquad (1)$$

Each register has one clocking bit, which is $R_1[8]$, $R_2[10]$ and $R_3[10]$, respectively. At each step, the majority value m of the three clocking bits is calculated, and only the registers whose clocking bit equals m are shifted. Therefore, two or three registers are shifted at each step. The output bit is the XOR of the most significant bits, $R_1[18] \oplus R_2[21] \oplus R_3[22]$. Figure 1 shows the structure of A5/1.

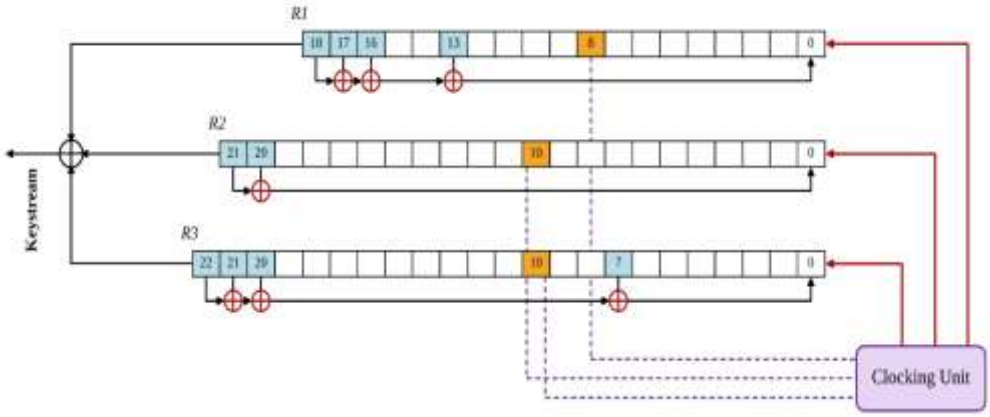

Figure 1: The structure of A5/1. The orange cells are the clocking bits.

The original initialization consists of four steps:

1. All registers are set to zero.
2. For 64 cycles, all registers are clocked regularly (the majority rule is ignored), and key bit k_i is XORed into the least significant bit of each register.
3. For 22 cycles, the frame number (FN) is loaded in the same way.
4. The registers are clocked 100 times using the majority rule, and the output is discarded.

The next 228 output bits form the keystream for one TDMA frame of 4.615 ms (114 bits for each direction). Since steps 2 and 3 are linear over GF(2), the state after step 3 is $L(K) \oplus L'(\mathrm{FN})$, where L and L' are linear maps.

4. The Four 128-bit Key Loading Techniques

In all four techniques, the A5/1 registers, feedback polynomials, majority clocking and output function are kept unchanged, and only the initialization is modified. The 128-bit key is denoted by $K = (k_0, \ldots, k_{127})$. When a technique splits the key, $K_A = (k_0, \ldots, k_{63})$ and $K_B = (k_{64}, \ldots, k_{127})$, with $K_B[i] = k_{64+i}$. The term "clock all" refers to regular clocking of the three registers, while "clock maj" refers to majority clocking. Algorithm 1 presents the four procedures, and Table 1 summarizes them.

Direct extension (D). In this technique, the 128 key bits are loaded in the same way as the original 64 bits, over 128 regular clock cycles. The loading of the frame number and the 100 warm-up cycles are not changed.

Extended warm-up with key salt (WS). K_A is loaded linearly, and then the frame number is loaded. The warm-up period is extended from 100 to 10,000 majority-clocked cycles. During the first 64 cycles of the warm-up, bit $K_B[i]$ is XORed into R_3 as a "salt". Since the salt is added after majority clocking has started, K_B enters through a non-linear state update.

Non-linear key scrambling (SC). K_A is loaded over 64 cycles as in the original cipher. In the same cycle i, the bit $s_i = K_A[i] \wedge K_B[i]$ is also XORed into R_1 and R_2. After that, the frame number is loaded and the standard 100-cycle warm-up is performed.

State mixing (SX). K_A and the frame number are loaded as in the original cipher. In each of the 100 warm-up cycles i, after the majority clock, a state-mix bit is computed as in Eq. (2)

$$x_i = (R_1[8] \wedge R_2[10]) \oplus K_B[i \bmod 64] \qquad (2)$$

and XORed into the three registers. In this way, K_B enters through the non-linear, state-dependent warm-up, and the two clocking bits are fed back into the state through a non-linear AND term. Since the warm-up has 100 cycles and K_B has 64 bits, bits $K_B[0]$ to $K_B[35]$ are injected again in cycles 64 to 99, so every bit of K_B is used at least once.

Table 1: Summary of the key-loading techniques. Setup cycles count the clock cycles before the first keystream bit.

Technique	K_B enters	Operation	vs. FN	Setup cycles
Original	— (64-bit key)	linear	—	64 + 22 + 100 = 186
D	linear loading	XOR	before	128 + 22 + 100 = 250
WS	warm-up, R_3	XOR, majority-clocked	after	64 + 22 + 10,000 = 10,086
SC	loading, R_1, R_2	AND with K_A, XOR	before	64 + 22 + 100 = 186
SX	warm-up, all	XOR with AND term	after	64 + 22 + 100 = 186

5. Experimental Setup

Implementation. All variants were implemented in C language. The baseline implementation was checked against the reference test vector (Section 3), and the SX implementation was cross-checked with the original thesis code of the first author.

Statistical testing. The NIST Statistical Test Suite, SP 800-22 Rev. 1a (Bassham et al., 2010), was used to evaluate the randomness of the keystreams. For each configuration, a keystream of 10^8 bits was generated under a fixed 128-bit key by concatenating the 228-bit outputs of consecutive frames, starting from frame number 0. The keystream was divided into $m = 100$ sequences of $n = 10^6$ bits, and the tests were run with the default parameters and a significance level of $\alpha = 0.01$. NIST defines two second-level criteria for a set of sequences (Bassham et al., 2010):

1. *Proportion.* The proportion of sequences with P-value $\geq \alpha$ must lie in the interval $(1 - \alpha) \pm 3\sqrt{(\alpha(1 - \alpha)/m)}$. For $m = 100$, at least 96 sequences must pass. For the random-excursion tests, the bound depends on the number of eligible sequences.
2. *Uniformity.* The distribution of the 100 P-values is checked using a χ^2 test over ten bins, which gives P-value$_T$. Uniformity is accepted if P-value$_T \geq 0.0001$.

A test is passed only if both criteria are satisfied. Under the null hypothesis, P-values are uniformly distributed on [0, 1]. Therefore, a higher P-value does not mean "more randomness", and configurations should not be ranked by their P-values (Sýs et al., 2015).

6. Results

6.1 NIST statistical tests

Table 2 shows the uniformity values P-value for the original cipher and the four techniques, and Figure 2 presents them on a logarithmic scale against the 0.0001 threshold. For the tests that have many subtests (Non-overlapping Template, Random Excursions and Random Excursions Variant), the table shows the mean over the subtests for compactness only, while the pass decision was made on each of the 188 subtests. All proportions were within the NIST bound.

It can be seen from Table 2 that all entries are above 0.0001. The smallest value, 0.0043 for the second Cumulative Sums test under D, is still more than 40 times larger than the threshold. Consequently, all four key-loading techniques pass every subtest of the suite, and the original cipher passes all subtests except the borderline case mentioned above.

These results are expected. In one full NIST run there are about 188 uniformity values, so some values below 0.05 or even 0.01 can occur by chance for a perfect generator (Sýs et al., 2015). The small value for D is consistent with chance and does not indicate a statistical defect. Similarly, the high values of some techniques (for example, 0.955835 for Approximate Entropy under SX) do not indicate better randomness. The statistical tests show that none of the key-loading modifications introduced a detectable bias, but they cannot show which technique is more secure.

Table 2: NIST SP 800-22 uniformity values P-value$_T$ (100 sequences of 10^6 bits). A test passes if P-value$_T \geq$ 0.0001 and the proportion of passing sequences is within the NIST bound.

#	Test	Original	D	WS	SC	SX
1	Frequency	0.334538	0.574903	0.759756	0.759756	0.162606
2	Block Frequency	0.994250	0.616305	0.798139	0.616305	0.554420
3	Cumulative Sums (fwd)	0.419021	0.595549	0.779188	0.115387	0.137282
4	Cumulative Sums (rev)	0.699313	0.004301	0.834308	0.719747	0.719747
5	Runs	0.996335	0.595549	0.040108	0.955835	0.514124
6	Longest Run	0.779188	0.955835	0.474986	0.991468	0.304126
7	Rank	0.699313	0.574903	0.129620	0.334538	0.071177
8	DFT (Spectral)	0.834308	0.514124	0.102526	0.699313	0.236810
9	Non-overlapping Template	0.546735	0.452202	0.521357	0.490807	0.515270
10	Overlapping Template	0.637119	0.181557	0.289667	0.798139	0.191687
11	Universal	0.129620	0.262249	0.289667	0.401199	0.911413
12	Approximate Entropy	0.020548	0.911413	0.045675	0.699313	0.955835
13	Random Excursions	0.427969	0.367128	0.461022	0.421607	0.782337
14	Random Exc. Variant	0.334525	0.390950	0.441300	0.342604	0.501042
15	Serial 1	0.699313	0.334538	0.026948	0.616305	0.574903
16	Serial 2	0.401199	0.202268	0.171867	0.191687	0.366918
17	Linear Complexity	0.595549	0.334538	0.816537	0.474986	0.759756

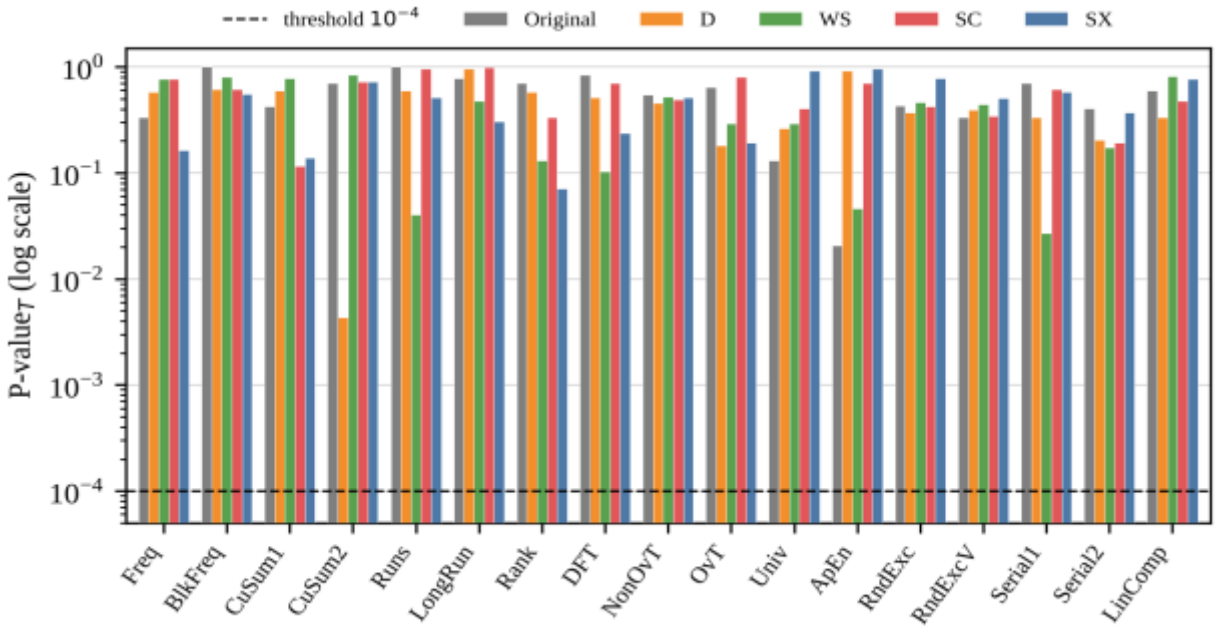

Figure 2: The uniformity P-values of all configurations against the NIST acceptance threshold (dashed line).

6.2 Structural analysis: effective key space

A general bound. If a technique absorbs the whole key into the 64-bit state before the frame number is loaded, every subsequent step depends only on that 64-bit state S_K and on the public frame number. In this case, the keystream of every frame is a function $F(S_K, \mathrm{FN})$, at most 2^{64} different keys can be distinguished, and an attacker can search S_K directly in 2^{64} steps, regardless of the nominal key length. As shown in Table 1, D and SC have this property. In contrast, WS and SX inject K_B only after the frame number has been loaded, so they are not subject to this bound.

Direct extension. Under D, the map $K \mapsto S_K$ is linear over GF(2) from $\{0, 1\}^{128}$ to $\{0, 1\}^{64}$. Gaussian elimination shows that its rank is 64, so its kernel has dimension 64. Therefore, every key belongs to a class of 2^{64} keys that produce exactly the same keystream on every frame. Using a kernel vector, the following pair of keys was obtained, which differs in 27 bit positions (k_0 is the least significant bit):

$$\mathrm{K} = \mathrm{CD613E30D8F16ADF\ 91B7584A2265B1F5}$$
$$\mathrm{K}' = \mathrm{CD613E30D8F16ADE\ 60B5236A4CABF0F4}$$

The two keys produce identical 228-bit keystreams on frame 0x134 and on 20 other random frames.

Non-linear scrambling. Under SC, bit $K_B[i]$ affects the state only through $s_i = K_A[i] \wedge K_B[i]$. When $K_A[i] = 0$, the value of $K_B[i]$ has no effect. As a result, about half of the bits of K_B are ignored for a random key. For the key K above, 33 of the 64 bits are ignored, and flipping one of these bits leaves the keystream unchanged on all 20 tested frames. Moreover, SC is also subject to the general 2^{64} bound.

WS and SX. In both techniques, K_B is XORed into the state during the majority-clocked warm-up, after the frame number has been loaded. This step is not linear in the key, and the key cannot be compressed into a 64-bit value that is independent of the frame. No equivalent keys were found for these two techniques. Therefore, exhaustive search over 128 bits is the generic attack on the key; no shortcut was found in this work, although one cannot be ruled out (Section 7).

6.3 Initialization cost and throughput

Table 3 and Figure 3 show the cost of each technique. SC and SX keep the original 186 setup cycles; SX adds one AND gate and one XOR gate per cycle, while D adds 64 cycles. In software, D, SC and SX run at 93–99% of the original frame rate. In contrast, WS needs 54 times more setup cycles and achieves only 3.6% of the original throughput. In GSM, a new initialization is needed for every 4.615 ms frame, which corresponds to about 217 frames per second in each direction; therefore, all variants are much faster than real time in software. However, the overhead of WS would be significant in hardware, where initialization latency and energy per frame are the main costs.

Table 3: Initialization cost and software throughput (frames/s including key setup and 228 output bits; median of 7 runs).

Technique	Setup cycles	Cycles per frame	Frames/s	Relative
Original	186	414	334,080	1.00
D	250	478	317,551	0.95
WS	10,086	10,314	11,952	0.04
SC	186	414	329,589	0.99
SX	186	414	310,940	0.93

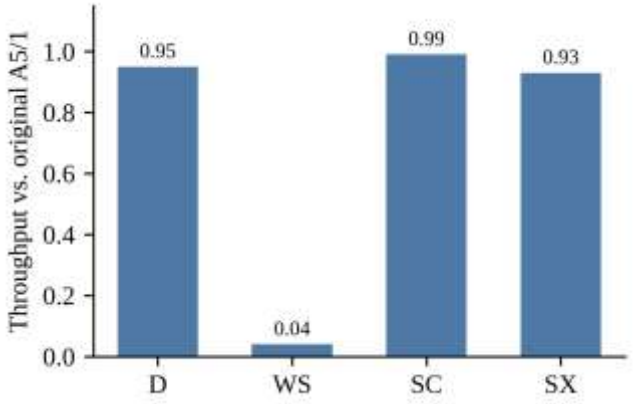

Figure 3: Software throughput of each technique relative to the original cipher.

7. Discussion and Limitations

Comparison. Table 4 combine the results of the three evaluations. Since all four techniques produce good randomness of keystreams, the choice between them must depend on structure and cost. D and SC appear to be 128-bit designs, but they provide no more than 264 security against key search, because the whole key passes through the 64-bit state before the frame number is loaded. Therefore, they offer no improvement over the original cipher. WS uses the full key, but it increases the initialization cost by a factor of 54. SX uses the full key, keeps the original number of setup cycles, adds only one AND gate and one XOR gate per cycle, and retains 93% of the software throughput. Consequently, SX is the recommended technique.

Table 4: Overall comparison of the four techniques.

	D	WS	SC	SX
Passes NIST SP 800-22	yes	yes	yes	yes
All 128 key bits affect the keystream	no (2^{64} classes)	yes	no (≈32 bits ignored)	yes
Key-search cost	$\leq 2^{64}$	2^{128}	$\leq 2^{64}$	2^{128}
Setup cycles	250	10,086	186	186
Relative throughput	0.95	0.04	0.99	0.93

The NIST suite is necessary but not sufficient. D passed every NIST test, although it has 2^{64} equivalent keys per class. This result shows that a key-extension proposal cannot be validated by randomness tests alone, and that ranking proposals by P-values is not meaningful. Therefore, structural checks of the key-loading map, together with equivalent-key tests, should be part of the standard evaluation.

Limitations. This study has four limitations:

1. **The internal state is still 64 bits**. The keystream generation phase of all variants is identical to that of A5/1. Therefore, state-recovery attacks on a single frame, such as TMDTO and guess-and-determine attacks (Biham & Dunkelman, 2000; Biryukov et al., 2001; Mahalanobis & Shah, 2014), still apply. According to the TMDTO rule (Babbage, 1995; Biryukov & Shamir, 2000), a 128-bit key requires at least 256 bits of state. A 128-bit key increases the cost of exhaustive key search under SX and WS, but it does not increase the cost of attacks on the state. Furthermore, an attacker who recovers the state of one frame could try to run the SX warm-up backwards by guessing bits of K_B. This attack has not been analyzed in this paper, and no more than 64-bit security is claimed against it.
2. **No formal cryptanalysis**. Formal cryptanalysis (algebraic, correlation or differential) of SX has not been carried out. The absence of equivalent keys in the experiments is not a proof that none exist.

3. **Software measurements only**. The performance was measured in software only. Hardware area, latency and energy are left for future work.
4. **Not compatible with GSM**. The described techniques are not suitable for use with deployed GSM terminals. They are rather designed for systems that make use of A5/1-like hardware. As for new designs, NIST lightweight standard candidate Ascon (Turan et al., 2025) and modern ciphers like ZUC-256 (Li et al., 2023) are preferable.

8. Conclusion

In this paper, four techniques for loading a 128-bit key into the unchanged A5/1 register structure were compared. All four techniques passed the complete NIST SP 800-22 suite; therefore, statistical testing could not distinguish between them. However, the structural analysis could. Direct extension and non-linear scrambling compress the key into the 64-bit state before the frame number is loaded, which leaves them with at most 2^{64} effective keys. This was confirmed by an explicit pair of equivalent keys for D and by ignored key bits for SC. Extended warm-up with salt and state mixing both use the full key, but only state mixing achieves this at the original initialization cost (186 cycles and 93% throughput). Therefore, state mixing is recommended as a low-cost key-loading upgrade. Future work will focus on enlarging the internal state to meet the TMDTO bound, analyzing SX against backtracking, algebraic and correlation attacks, and measuring its area and energy on an FPGA.

Data Availability

The C source code of all variants and test programs is available from the corresponding author upon request.

Declaration on the Use of AI Tools

During the preparation of this manuscript, the authors used an AI language model to assist with language editing and document formatting. The authors reviewed and verified all content, and take full responsibility for the work.

References

Ahmad, F., Khan, R., & Nkembi, A. A. (2025). Improvement in the interception vulnerability level of encryption mechanism in GSM. *Inventions, 10*(4), 56.

Babbage, S. (1995). Improved exhaustive search attacks on stream ciphers. In *European Convention on Security and Detection* (IEE Conference Publication No. 408, pp. 161–166). IEE.

Bassham, L. E., Rukhin, A. L., Soto, J., Nechvatal, J. R., Smid, M. E., Leigh, S. D., Levenson, M., Vangel, M., Heckert, N. A., & Banks, D. L. (2010). *A statistical test suite for random and pseudorandom number generators for cryptographic applications* (NIST Special Publication 800-22 Rev. 1a). National Institute of Standards and Technology.

Biham, E., & Dunkelman, O. (2000). Cryptanalysis of the A5/1 GSM stream cipher. In *Progress in Cryptology – INDOCRYPT 2000* (LNCS 1977, pp. 43–51). Springer.

Biryukov, A., & Shamir, A. (2000). Cryptanalytic time/memory/data tradeoffs for stream ciphers. In *Advances in Cryptology – ASIACRYPT 2000* (LNCS 1976, pp. 1–13). Springer.

Biryukov, A., Shamir, A., & Wagner, D. (2001). Real time cryptanalysis of A5/1 on a PC. In *Fast Software Encryption – FSE 2000* (LNCS 1978, pp. 1–18). Springer.

Briceno, M., Goldberg, I., & Wagner, D. (1999). *A pedagogical implementation of the GSM A5/1 and A5/2 voice privacy encryption algorithms*.

Fauzi, S., Othman, M., Shuib, F., Seman, K., & Rahim, K. (2018). Randomness evaluation of modified A5/1 stream cipher for global system for mobile communication. *Malaysian Journal of Science Health & Technology, 2*.

Golić, J. D. (1997). Cryptanalysis of alleged A5 stream cipher. In *Advances in Cryptology – EUROCRYPT '97* (LNCS 1233, pp. 239–255). Springer.

Jawad, N. H., Katran, L. F., Albermany, S., & Bani, S. J. (2024). Enhancement A5/1 stream cipher algorithm as lightweight generator. *AIP Conference Proceedings, 3092*, 070013.

Joshi, A., & Mishra, A. (2021). Enhanced A5 algorithm for secure audio encryption. In *Proceedings* (pp. 163–174). Springer Singapore.

Jurecek, M., Bucek, J., & Lorencz, R. (2019). Side-channel attack on the A5/1 stream cipher. In *Proceedings of the 22nd Euromicro Conference on Digital System Design (DSD)* (pp. 633–638). IEEE.

Koch, M., Nettersheim, C., Horstmann, T., & Rademacher, M. (2025). *A5/1 is in the air: Passive detection of 2G (GSM) ciphering algorithms* [Preprint]. arXiv.

Li, C., Chen, J., Wang, L., & Shu, Y. (2023). Design of ZUC-256 stream cipher coprocessor for 5G communication security. In *Proceedings of the 3rd International Symposium on Computer Technology and Information Science (ISCTIS)* (pp. 1122–1127). IEEE.

Mahalanobis, A., & Shah, J. (2014). An improved guess-and-determine attack on the A5/1 stream cipher. *Computer and Information Science, 7*(1).

Mammeri, Z. (2024). *Cryptography: Algorithms, protocols, and standards for computer security*. Wiley.

Morgan, D. (2025). Determining the A5 encryption algorithms used in 2G (GSM) networks (pp. 45–61). Springer Nature Switzerland.

Prajapat, R. P., Bhadada, R., & Choudhary, A. (2022). Information security enhancement by increasing randomness of stream ciphers in GSM. *International Journal of Advanced Computer Science and Applications, 13*(4).

Stallings, W. (2017). *Cryptography and network security: Principles and practice* (7th ed.). Pearson.

Sýs, M., Říha, Z., Matyáš, V., Márton, K., & Suciu, A. (2015). On the interpretation of results from the NIST statistical test suite. *Romanian Journal of Information Science and Technology, 18*(1), 18–32.

Turan, M. S., McKay, K., Chang, D., Kang, J., & Kelsey, J. (2025). *Ascon-based lightweight cryptography standards for constrained devices* (NIST Special Publication 800-232). National Institute of Standards and Technology.