The DRACO Stream Cipher

Fast Software Encryption 2023

Matthias Hamann, ERNW Research GmbH, Heidelberg,
Alexander Moch, Universität Mannheim,
Matthias Krause, Universität Mannheim,
Vasily Mikhalev, Universität Siegen.

20 March 2023

Kobe, Japan

Motivation for small-state stream ciphers

What is the problem with stream ciphers?

- Time-memory-data tradeoff attacks limit the security level to half the state size.
- Taken from `https://www.ecrypt.eu.org/stream/call/`:
  - Time-memory trade-offs mean that the size of the stream cipher state should be appropriate for the claimed security level. For instance, the state size should be at least twice the key size.
  - Any key-recovery attack (including time-memory-data tradeoff attacks) should be at least as difficult as exhaustive search.
  - Also, distinguishing attacks are likely to be of interest to the cryptographic community. However the relative importance of high complexity distinguishing attacks may become an issue for wider discussion.
- State is typically twice the key size.

Why small-state ciphers?

What is the purpose of reducing the state size?

- State cells (i.e., flip-flops) induce substantial hardware cost in terms of area and power consumption!
- Reduced hardware costs can mean:
  - Reduced monetary costs, even few cents matter for RFID tags (produced in the millions/billions).
  - But also a widening of application scenarios, e.g., due to reduced power/energy consumption.

Area

How do we measure hardware costs?

- Typically measured in gate equivalents (GEs) to allow for comparing designs/implementations over different standard cell libraries.
- 1 GE corresponds to the area of a two-input drive-strength-one NAND gate.
- Chip area strongly influences the manufacturing costs of RFID tags.
- Still, focus has shifted towards power and energy consumption over the recent years.

Cell	NOT	NAND	NOR	AND	OR	MUX	XOR	Flip-Flop
Area	0.67	1.00	1.00	1.33	1.33	2.33	2.67	5.33 – 12.33

Power vs. Energy

Power $\neq$ Energy

- Low-cost RFID tags are usually passively powered (i.e., via an electromagnetic field radiated by the reader).
- Power consumption is crucial, e.g., it ...
  - ... determines whether design/implementation works at all in the field.
  - ... limits the range of passive RFID tags in the field.
  - ... affects the temperature emitted by the circuit (medical implants!).
- Energy consumption relevant, e.g., for actively powered RFID tags (battery life of a pace maker!).

Static vs. Dynamic Power Consumption

Static power consumption:

- Caused by leakage currents.
- E.g., power consumption of a flip-flop holding a constant value.

Dynamic power consumption:

- Caused by switching activity, i.e., signal changes.
- E.g., additional power consumption of a flip-flop due to a change of its stored value.

Already for moderate clock frequencies (here: between 10 MHz and 20 MHz) dynamic power consumption dominates static power consumption of flip-flop storage cells.

- To the best of our knowledge, DRACO is the first cipher design to use this effect intentionally.

Existing small-state stream ciphers

What ciphers exist in this field?

- Small-state ciphers divide the state into a volatile and a non-volatile part.
  - The non-volatile part is typically some external resource, outside of the cipher hardware module.
- Continuous Key: Sprout, Plantlet, Fruit, Atom.
  - All vulnerable to a distinguishing attack limiting the security to half of the volatile state size.
- Continuous IV: None.
  - Suggestion with proof at SAC 2019.

Our Contribution

What do we do?

- Improve upon the bound of the SAC 2019 work.
    - In the non-volatile state, use a *key prefix* in addition to the IV.
    - We use basically the same model as SAC 2019.
        - Slightly modified for an easier proof.
        - The TMDTO attacks for all ciphers work in this model.
        - The lower bounds match the upper bounds.
- We introduce a *packet*.
- Present a stream cipher called DRACO built upon this.
    - Small-state cipher with a modular design.
    - 128-bit secret key and 128-bit *volatile* internal state.
    - Using a 96-bit non-volatile IV and a non-volatile 32-bit key prefix.

Abstracting stream ciphers

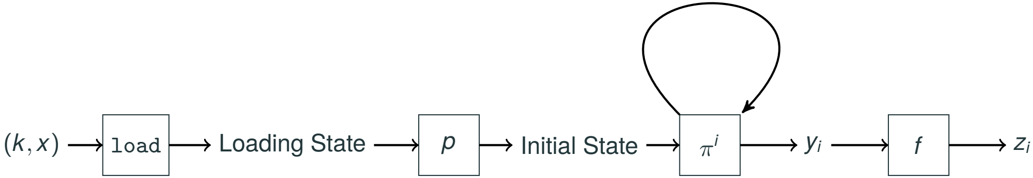
How can we abstract stream ciphers?

Loading: load the key and the IV into the hardware module.

Initialisation: mix/permute the (volatile) internal state.

State update: update one state to the next.

Output: create an output bit from the current internal state.



Packet mode

What is this?

- We limit the amount of keystream bits per key-IV combination.
- Afterwards a new IV must be used.
- Two packets with different IVs share *no* internal state.

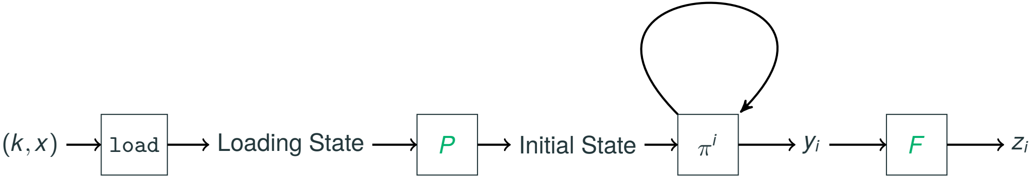
Why is this reasonable?

- A5/1 produces only 228 keystream bits per key/IV pair.
- Bluetooth packets contain at most 2790 bits for the so-called basic rate.
- For wireless local area networks (WLANs) at most 11454 bytes (i.e., $< 2^{17}$ bits) are encrypted under the same key/IV pair using CCMP.
- For TLS 1.3, the maximum amount of data encrypted under the same key/IV pair is $2^{14} + 2^{8}$ bytes (i.e., $2^{17} + 2^{11} < 2^{18}$ bits).

The model

How do we prove security?

- Basically replace the mixing and the output function by random oracles.
- Initialisation is replaced by a random permutation.
    - Only volatile part is permuted.
    - Pick a new independent random permutation for each non-volatile state.
- The output function is replaced by a random function.
    - Pretty strong, but attacks still work.



(k, x) → load → Loading State → P → Initial State → π^i → y_i → F → z_i

The proof

What are the adversary's capabilities?

- Query the construction, as well as P and F.
- Receive internal states and secret key after the queries.
- Queries, internal states and secret key are collected in a transcript.

What is the bound?

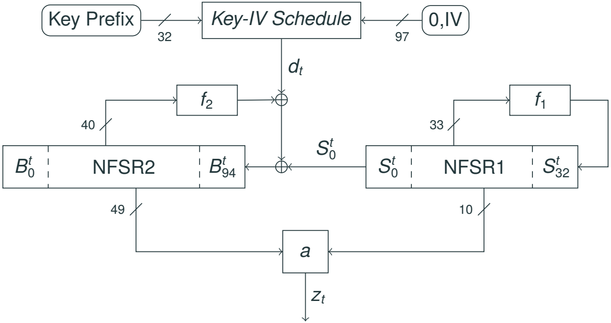
- We are using the H-coefficients technique.
- The final bound is

$$\underset{\mathbf{A}}{\Delta}\left(\mathcal{O}_{\text{real}}, \mathcal{O}_{\text{ideal}}\right) \leq \frac{q/2}{2^{\ell_k}} + \frac{q/2}{2^{\ell_v} - q}.$$

DRACO

What does it look like?

- Grain-like structure.
- 128-bit secret key.
- 128-bit volatile internal state.
- 128-bit non-volatile state.
    - 96-bit IV & 32-bit key prefix.
- Two NFSRs:
    - One based on Achterbahn with the maximum period of 2^{33}.
    - The other based on Grain-128a.
- State initialisation for 512 cycles.
- Packet length of up to 2^{32} bits per key-IV pair.



Flexible design of DRACO

Our design approach allows for four implementation variants of Draco.

Regular: If the key prefix and the IV are both externally available.

K: If the IV is externally available.

I: If the key prefix is externally available.

KI: The key prefix and the IV are both held inside of the Draco hardware module.

Attack by Subhadeep Banik

Distinguishing attack with 108 bits complexity in the next talk!

What is the problem with DRACO?

- The combination of three properties:
  - Small key-prefix.
  - The period of the KIS-bit is small.
  - For some IVs the KIS-bit is zero for several cycles.
- *Quick fix*: extend the key-prefix to full 128 bits.

In clock cycle t the *key-IV-schedule bit* (KIS bit) d_t is computed as

$$d_t := \begin{cases} x_{t \bmod 97}, & \text{for } 0 \leq t \leq 255, \\ K_{t \bmod 32} \oplus x_{t \bmod 97}, & \text{for } t \geq 256, \end{cases}$$

where $x_0 := 0$ and $x_i := IV_{i-1}$ for $i = 1, \ldots, 96$.

Quick fix

How can we fix this?

- Extending the key prefix to the full length of 128 bits.
- Only marginally affects variants with an external key prefix.
- Worsens implementation variants with an internal key prefix significantly.

Hardware results

Design	**Area** [GE]	**Power** [µW]				
		100 KHz	1 MHz	10 MHz	100 MHz	1 GHz
Atom	2976	67.9	71.2	104.9	441.9	3811.7
Atom$_{[K]}$	3858	88.9	92.3	126.1	463.9	3842.3
Grain-128a	2795	67.3	71.6	115.3	551.4	4912.9
Draco	2142	48.8	51.6	79.2	355.6	3119.3
Draco$^{\text{QuickFix}}$	2334	52.0	54.8	83.5	370.3	3238.4
Draco$_{[K]}$	2368	54.2	57.0	84.7	369.1	3134.1
Draco$^{\text{QuickFix}}_{[K]}$	3215	73.0	75.9	104.6	392.3	3269.0
Draco$_{[I]}$	2805	64.6	67.7	95.1	372.3	3144.7
Draco$^{\text{QuickFix}}_{[I]}$	2997	67.8	70.6	99.4	387.0	3263.8
Draco$_{[KI]}$	3025	69.9	72.6	100.6	377.6	3150.0
Draco$^{\text{QuickFix}}_{[KI]}$	3872	88.7	91.5	120.3	408.0	3284.8

Back to a 32-bit key prefix (Work In Progress)

How can we go back to a 32-bit key prefix?

- The quick fix is acceptable for variants with an external key.
- With internal keys however, numbers got significantly worse.
- We want modularity.

In clock cycle t the *key-IV-schedule bit* (KIS bit) d_t is computed as

$$d_t := \begin{cases} x_{t \bmod 97}, & \text{for } 0 \leq t \leq 255, \\ K_{t \bmod 32} \oplus x_{t \bmod 97}, & \text{for } 256 \leq t \leq 511, \\ K_{t \bmod 15} \oplus K_{(t \bmod 17)+15} \oplus K_{\mathrm{addr(t)}} \oplus x_{t \bmod 97}, & \text{for } t \geq 512, \end{cases}$$

where $x_0 := 0$, $x_i := IV_{i-1}$ for $i = 1, \ldots, 96$, and

$$\mathrm{addr}(t) := (S^t_{28}, S^t_{14}, S^t_{22}, S^t_4, S^t_{24}) \in \{0, \ldots, 31\}.$$

Back to a 32-bit key prefix (Work In Progress)

Design	**Area** [GE]	**Power** [µW]				
		100 KHz	1 MHz	10 MHz	100 MHz	1 GHz
Atom	2976	67.9	71.2	104.9	441.9	3811.7
Atom$_{[K]}$	3858	88.9	92.3	126.1	463.9	3842.3
Grain-128a	2795	67.3	71.6	115.3	551.4	4912.9
Draco	2142	48.8	51.6	79.2	355.6	3119.3
Draco$^{\text{WIP}}$	2292	51.7	54.5	83.3	370.8	3245.9
Draco$_{[K]}$	2368	54.2	57.0	84.7	369.1	3134.1
Draco$^{\text{WIP}}_{[K]}$	2517	57.0	59.9	88.8	377.1	3260.8
Draco$_{[I]}$	2805	64.6	67.7	95.1	372.3	3144.7
Draco$^{\text{WIP}}_{[I]}$	2955	67.5	70.4	99.2	387.6	3271.3
Draco$_{[KI]}$	3025	69.9	72.6	100.6	377.6	3150.0
Draco$^{\text{WIP}}_{[KI]}$	3174	72.7	75.6	104.4	392.8	3276.6

Thank you for your attention.