Collision Attacks based on the Entropy Loss caused by Random Functions

Andrea Röck

WEWORC, 6th July 2007

INRIA

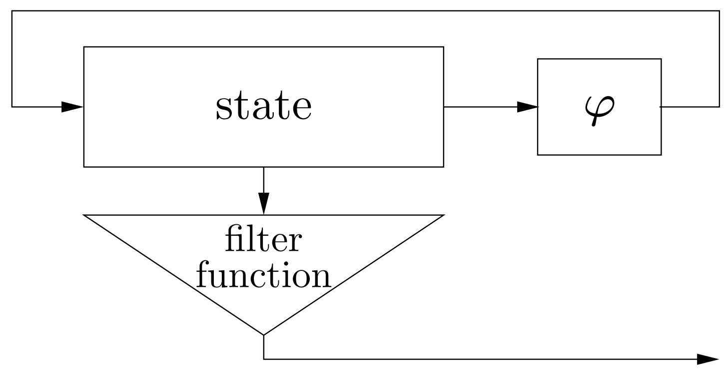
Outline

- Stream Cipher Model
- Entropy Estimation
  - Previous Results
  - New Entropy Estimator
- Collision Attacks
- Conclusion

Part 1

Stream Cipher Model

Stream Cipher Model



- State with values $s_k \in \Omega_n$ for all $k \geq 0$
  - State space $\Omega_n = \{\omega_1, \omega_2, \ldots \omega_n\}$
  - Initial state s_0
  - Initial distribution $\{p_i\}_{i=1}^{n}$ with $p_i = Pr[s_0 = \omega_i]$
- Update function $\varphi \in \mathcal{F}_n = \{\varphi : \Omega_n \to \Omega_n\}$

Our Stream Cipher Model

Assumption: Use *random function model* for the update function

- φ is randomly chosen out of $\mathcal{F}_n$
- All statistical statements are made on average over all $\varphi \in \mathcal{F}_n$

Motivation

- New stream cipher proposals which might fit into this model
  e.g. MICKEY (version 1) **[Babbage and Dodd 05]**
- The image of $\varphi^{(k)}$ is (much) smaller than n thus we *loose entropy*

Questions

- How much entropy do we loose in the state?
- Can this loss be efficiently exploited into a collision attack?

Notations

- Probability of a state being ω_i after k iterations of φ:

$$p_i^{\varphi}(k) = Pr[\varphi^{(k)}(s_0) = \omega_i]$$

- Entropy of the state after k iterations of φ:

$$H_k^{\varphi} = \sum_{i=1}^{n} p_i^{\varphi}(k) \log_2 \left(\frac{1}{p_i^{\varphi}(k)} \right)$$

- Expected entropy after k iterations taken over all $\varphi \in \mathcal{F}_n$:

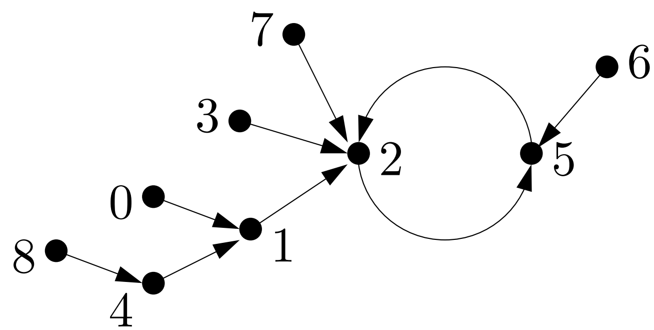
$$E(H_k)$$

Part 2

Entropy Estimation

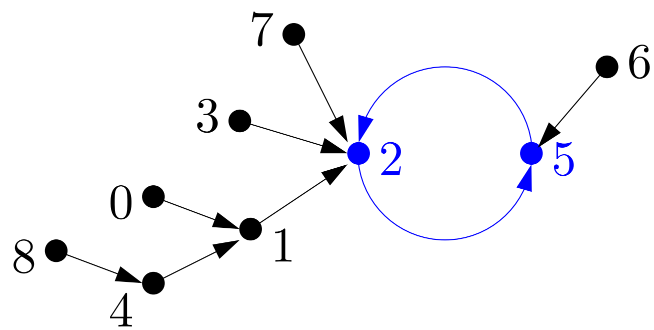
Previous Results

Properties of Random Functions



[Flajolet Odlyzko 90]

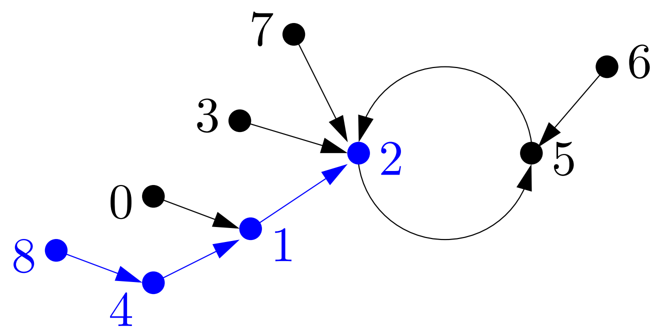
Properties of Random Functions



[Flajolet Odlyzko 90]

- # Cycle points: $CP(n) \quad = \sqrt{\pi\ n/2}$

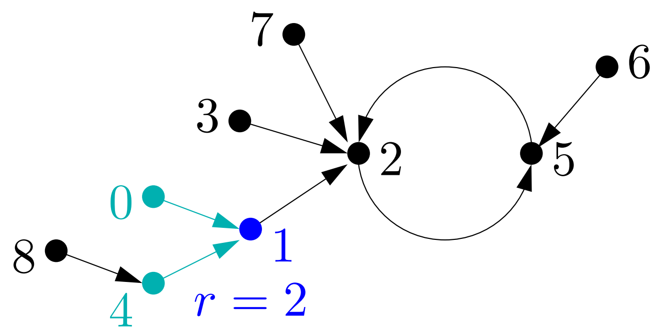
Properties of Random Functions



[Flajolet Odlyzko 90]

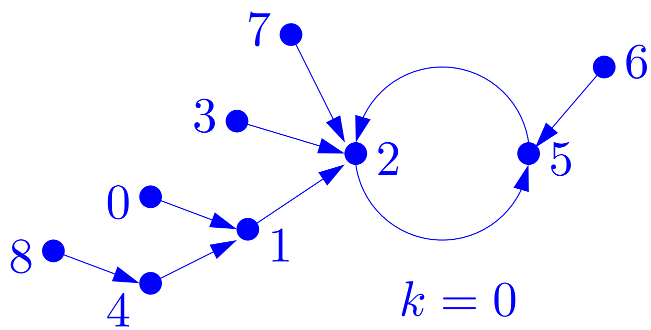
- # Cycle points: $CP(n) \quad = \sqrt{\pi \; n/2}$
- Maximal tail length: $MTL(n) = \sqrt{\pi \; n/8}$

Properties of Random Functions



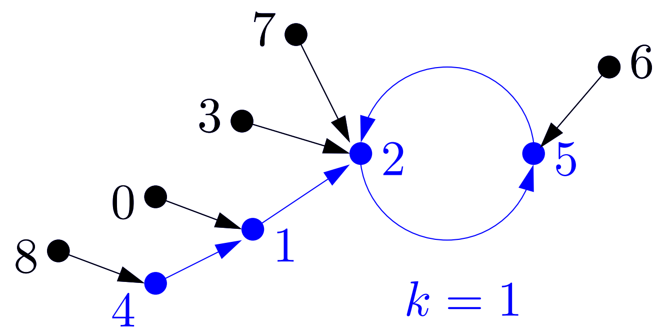
[Flajolet Odlyzko 90]

- # Cycle points: $CP(n) = \sqrt{\pi\, n/2}$
- Maximal tail length: $MTL(n) = \sqrt{\pi\, n/8}$
- # r-nodes: $RN(n, r) = n/r!e$



[Flajolet Odlyzko 90]

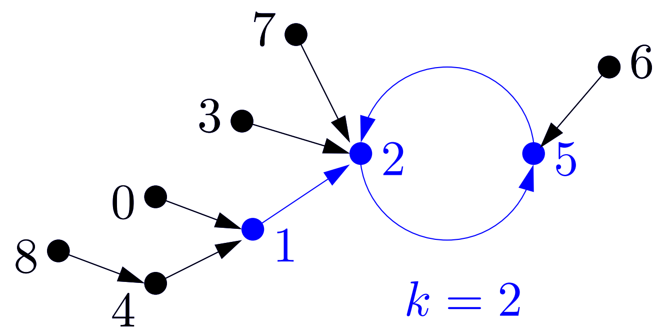
- # Cycle points: $CP(n) = \sqrt{\pi\, n/2}$
- Maximal tail length: $MTL(n) = \sqrt{\pi\, n/8}$
- # r-nodes: $RN(n,r) = n/r!e$
- # Image points: $IP(n,k) = (1 - \tau_k)\, n$
  where $\tau_0 = 0$ and $\tau_{k+1} = e^{-1+\tau_k}$



[Flajolet Odlyzko 90]

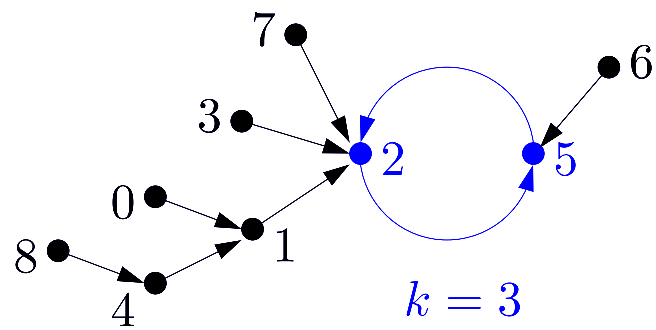
- # Cycle points: $CP(n) = \sqrt{\pi\, n/2}$
- Maximal tail length: $MTL(n) = \sqrt{\pi\, n/8}$
- # r-nodes: $RN(n, r) = n/r!e$
- # Image points: $IP(n, k) = (1 - \tau_k)\, n$
  where $\tau_0 = 0$ and $\tau_{k+1} = e^{-1+\tau_k}$

Properties of Random Functions



[Flajolet Odlyzko 90]

- \# Cycle points: $CP(n) = \sqrt{\pi\, n/2}$
- Maximal tail length: $MTL(n) = \sqrt{\pi\, n/8}$
- \# r-nodes: $RN(n,r) = n/r!e$
- \# Image points: $IP(n,k) = (1-\tau_k)\, n$
  where $\tau_0 = 0$ and $\tau_{k+1} = e^{-1+\tau_k}$



[Flajolet Odlyzko 90]

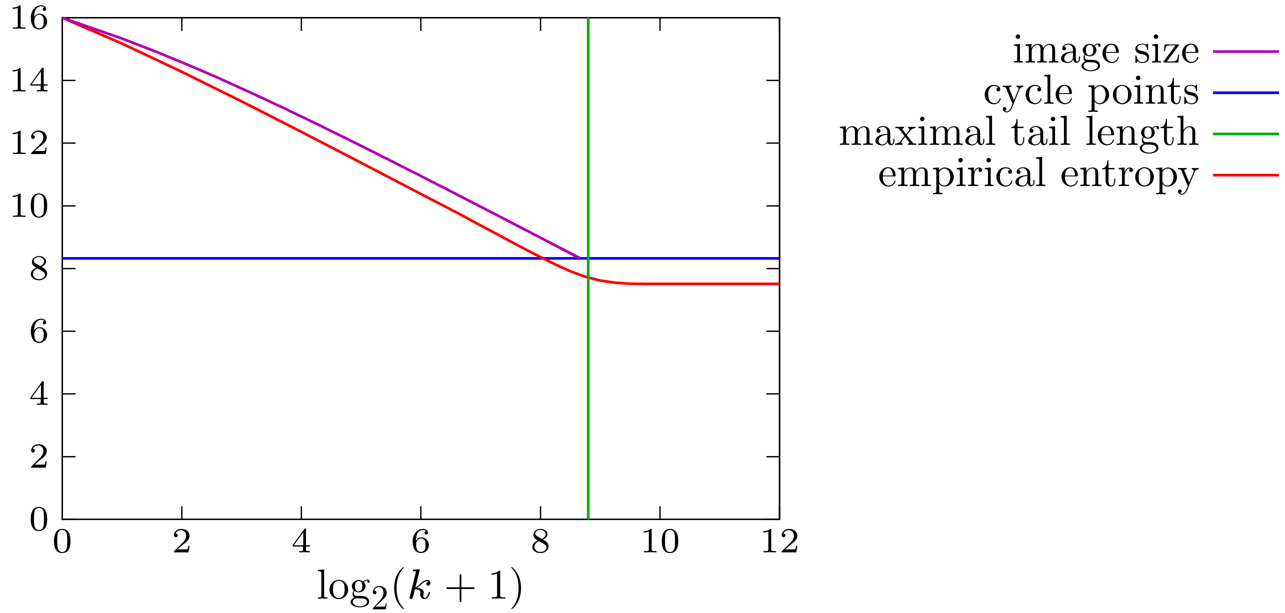
- \# Cycle points: $CP(n) \quad = \sqrt{\pi\, n/2}$
- Maximal tail length: $MTL(n) = \sqrt{\pi\, n/8}$
- \# r-nodes: $RN(n,r) = n/r!e$
- \# Image points: $IP(n,k) \ = (1 - \tau_k)\, n$
  where $\tau_0 = 0$ and $\tau_{k+1} = e^{-1+\tau_k}$

Bounding Entropy with Image Points [Hong Kim 05]

- Upper bound given by number of image points:

$$E(H_k) \leq log_2(n) + log_2(1 - \tau_k)$$

- Example for $n = 2^{16}$



New Entropy Estimator

New Entropy Estimator (1)

Motivation:

- Find an entropy estimator which is more precise than the upper bound given by the number of image points

Ideas:

- We assume a uniform initial distribution
- If a state can be produced by exactly r other states after one iteration, it has probability r/n

New Entropy Estimator (2)

- Average number of states produced by r states after k iterations:

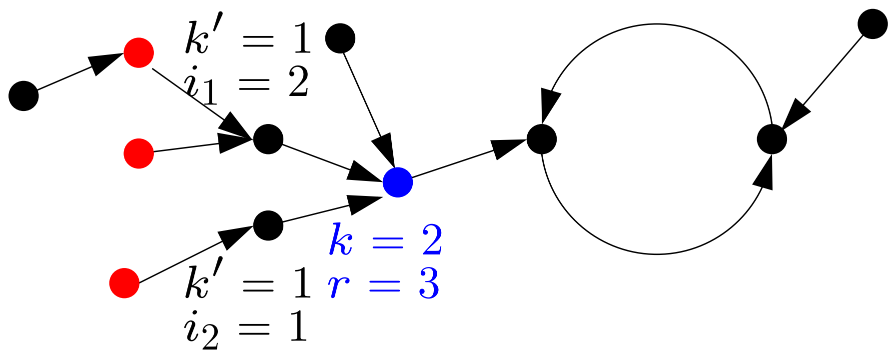
$$n\ c_k(r)$$

- Expected Entropy:

$$\begin{aligned} E(H_k) &= \sum_{r=1}^{n} n\ c_k(r)\ \frac{r}{n} \log_2 \frac{n}{r} \\ &\approx \log(n) - \sum_{r=1}^{n} c_k(r)\ r\ \ \log_2(r) \end{aligned}$$

Calculation of $c_k(r)$ (1)

- $k = 1$: Use directly $RN(n, k) = n/r!e$
- $k > 1$: Use the fact that such a tree node has
  - j children with $i_1, \ldots, i_j$ descendants after $k-1$ iterations, $i_1 + \cdots + i_j = r$, and
  - arbitrary tree children of depth $< k$



Calculation of $c_k(r)$ (2)

By analyzing the generating function of our property we get

$$c_k(r) = \frac{1}{e} f_1(k) S(k, r, 1)$$

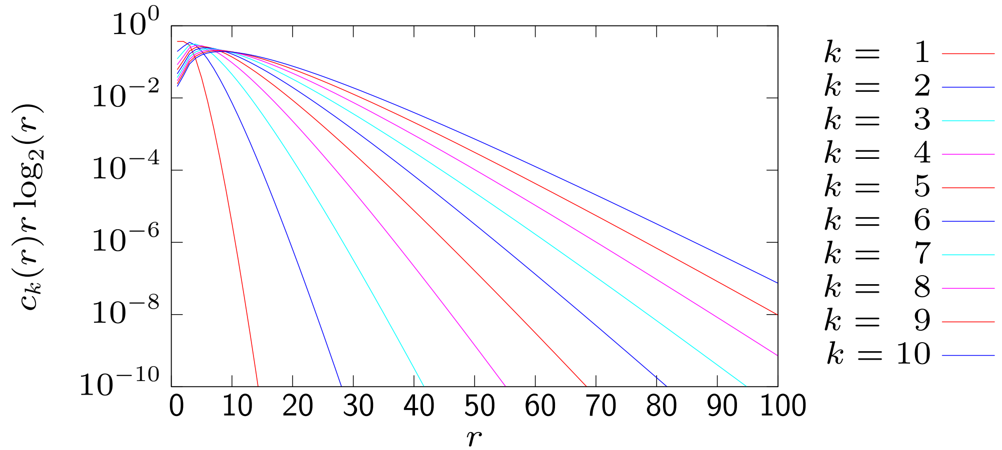
- $f_1(k) = e^{f_1(k-1)/e}$ with $f_1(1) = 1$
- $S(k, r, m) =$
$$\begin{cases} S(k,r,m) = \sum_{u=0}^{\lfloor \frac{r}{u} \rfloor} \frac{c_{k-1}(m)^u}{u!} S(k, r - mu, m+1) & \text{if } 1 \le m \le r \\ 1 & \text{if } r = 0 \\ 0 & \text{if } 0 < r < m \end{cases}$$

Remarks

- We ignore the incoming cycle nodes
- Complexity of computing $c_k(r)$ with $r \leq R$ and $k \leq K$:

$$O\left(K\ R^2 \log(R)\right)$$

- $E(H_k) = \log(n) - \sum_{r=1}^{R} c_k(r) r \log_2(r) - \sum_{r=R+1}^{n} c_k(r) r \log_2(r)$



Estimation of Entropy Loss with different Methods

k	0	1	2	3	...	7	8
empirical data $n = 2^{16}$	0.0000	0.8272	1.3456	1.7252	...	2.6690	2.8324
image points	0.0000	0.6617	1.0938	1.4186	...	2.2546	2.4032
$R = 30$	0.0000	0.8272	1.3457	1.7254	...	2.6561	2.8004
$R = 50$	0.0000	0.8272	1.3457	1.7254	...	2.6693	2.8324

- For small k our new estimator is more precise than the upper bound given by the number of image points
- For larger k we need a bigger R to have a small error

Part 3

Collision Attacks

Collision Attacks

Ideas:

- Using a random function leads to a loss of entropy
- A reduced entropy leads to higher probability of a collision
- If two states are the same, then the subsequent output sequences are identical
- Two proposals for an attack on MICKEY in **[Hong Kim 05]** (no real attacks)

Attack 1

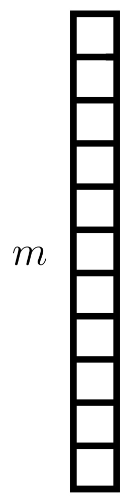
(Proposition [Hong Kim 05])

- Search for collision after k iterations

Attack 1

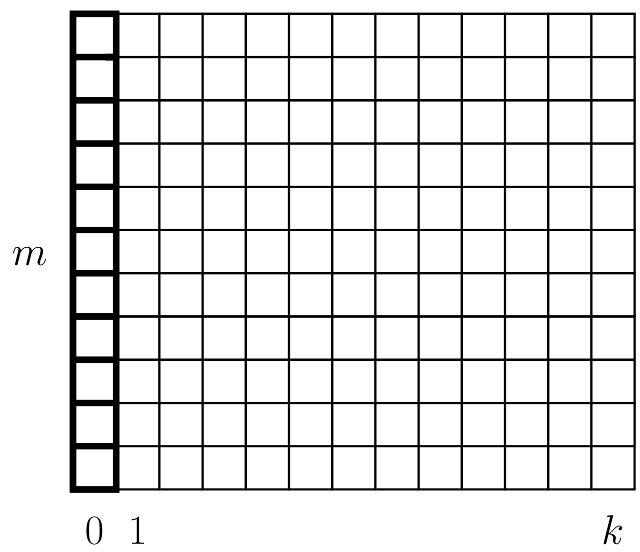
(Proposition [Hong Kim 05])

- Search for collision after k iterations



Attack 1
(Proposition [Hong Kim 05])

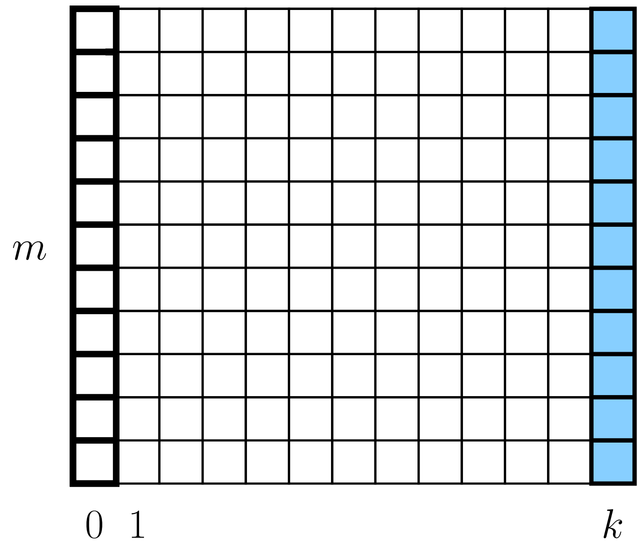
- Search for collision after k iterations



Attack 1

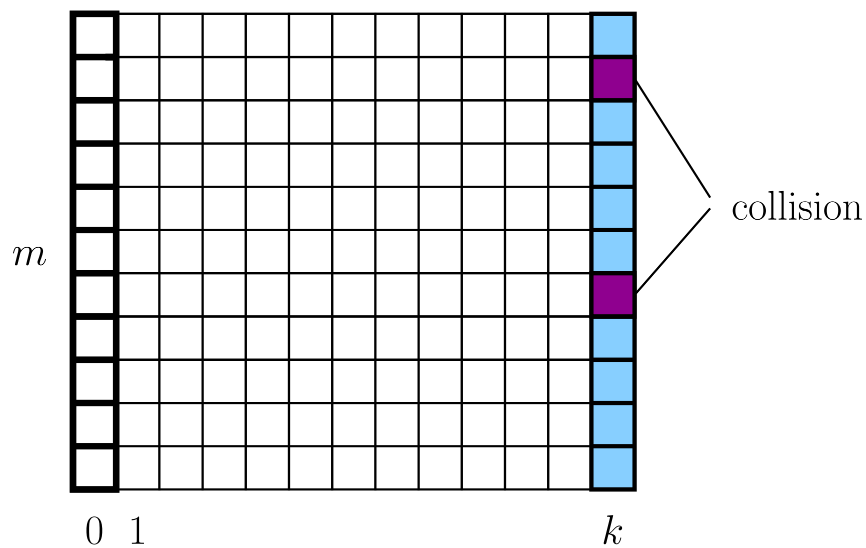
(Proposition [Hong Kim 05])

- Search for collision after k iterations



Attack 1
(Proposition [Hong Kim 05])

- Search for collision after k iterations



Attack 1

(Analysis (1))

- **Upper bound:** $E(H_k) \leq \log_2(n) - \log_2(k) + 1$
- **Birthday paradox:** Need $\sim \sqrt{\frac{n}{k}}$ values in the last row

	Attack 1
Space complexity **[Hong Kim 05]**	$\sim \sqrt{\frac{n}{k}}$
Data complexity **(new)**	$\sim \sqrt{k\ n}$

Attack 1
(Remark)

Under which circumstances is the attack effective?

- If we have functions which loose on average more than $2\log_2(k)$ bits after k iterations

This means that we don't use a random function, but the principle of the attack stays the same

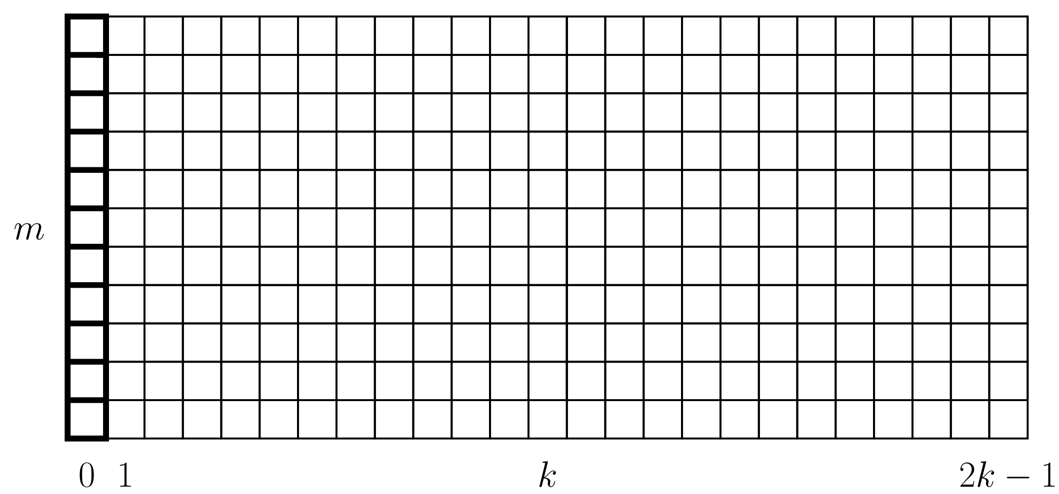
Attack 2

(Proposition [Hong Kim 05])

- Iterate $2k$ times and search for collision in the second half of the intermediate states

Attack 2
(Proposition [Hong Kim 05])

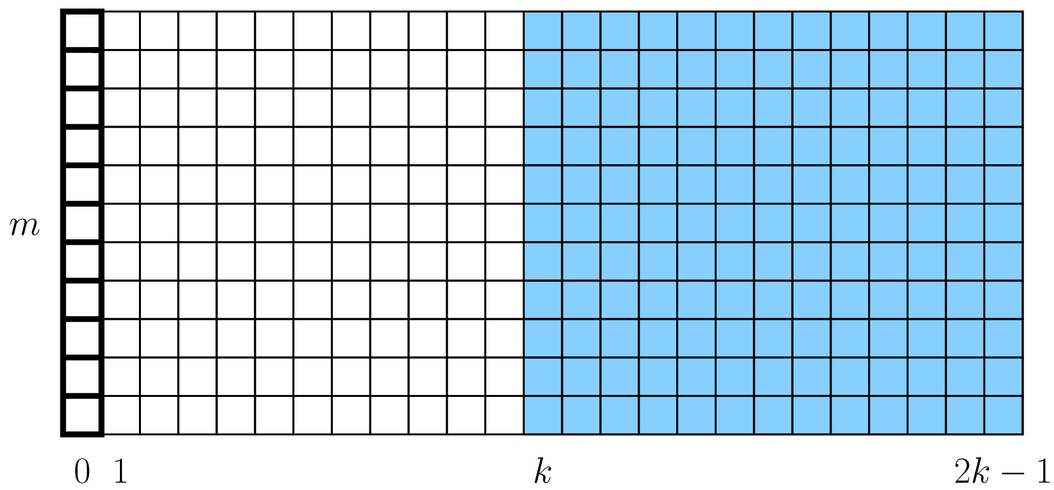
- Iterate $2k$ times and search for collision in the second half of the intermediate states



Attack 2

(Proposition [Hong Kim 05])

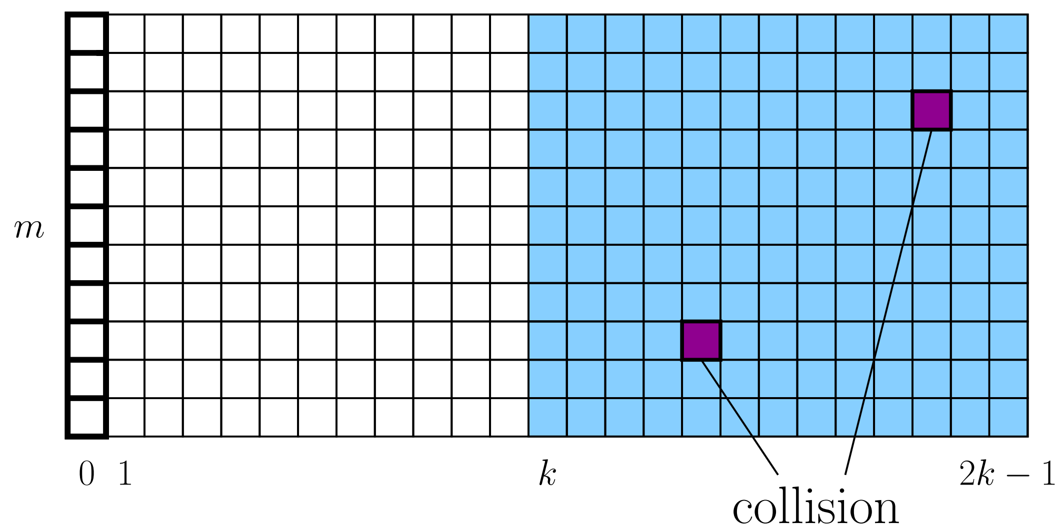
- Iterate $2k$ times and search for collision in the second half of the intermediate states



Attack 2

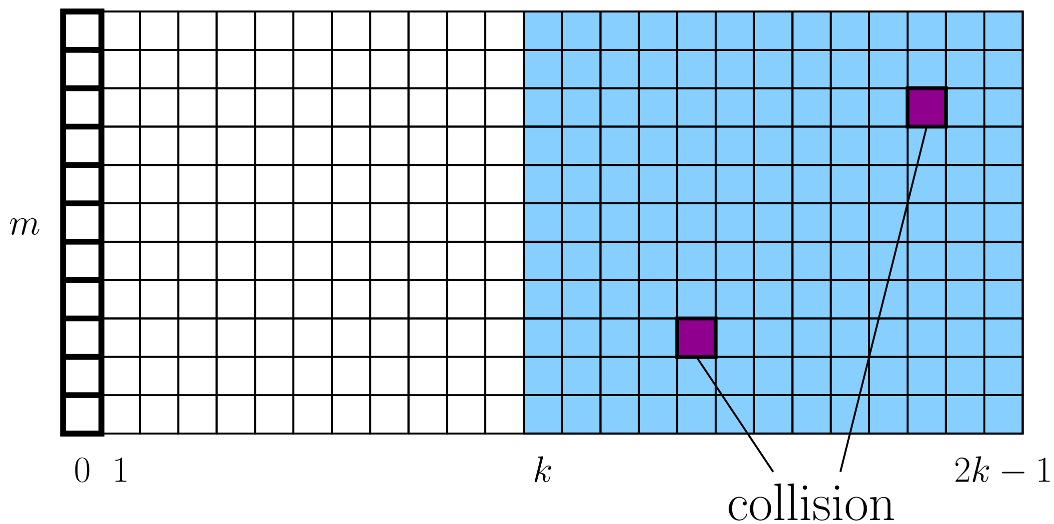
(Proposition [Hong Kim 05])

- Iterate $2k$ times and search for collision in the second half of the intermediate states



Attack 2
(Proposition [Hong Kim 05])

- Iterate $2k$ times and search for collision in the second half of the intermediate states



- **[Hong Kim 05]**: Magnitude of m such that $m\,k \sim \sqrt{n/k}$

Attack 2

(Analysis (new))

- Probability of collision is smaller than $1 - Pr[noColTotal]$
- By counting arguments we get:

$$Pr[noColTotal] = \frac{n(n-1)\cdots(n-2km+1)}{n^{2km}}$$

- **Birthday Paradox:** We need $2mk \approx \sqrt{n}$

	Attack 1	Attack 2
Space complexity	$\sim \sqrt{\frac{n}{k}}$	$\sim \sqrt{n}/2$
Data complexity	$\sim \sqrt{k\ n}$	$\sim \sqrt{n}$

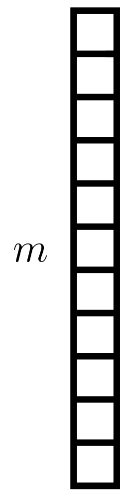
Attack 3 (new)
(Distinguished Points)

- Iterate until we reach a distinguished point

Attack 3 (new)

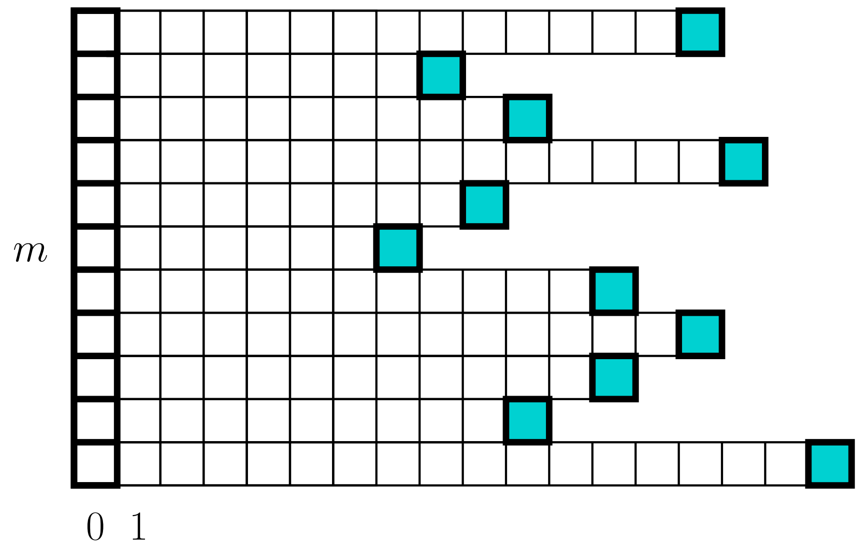
(Distinguished Points)

- Iterate until we reach a distinguished point



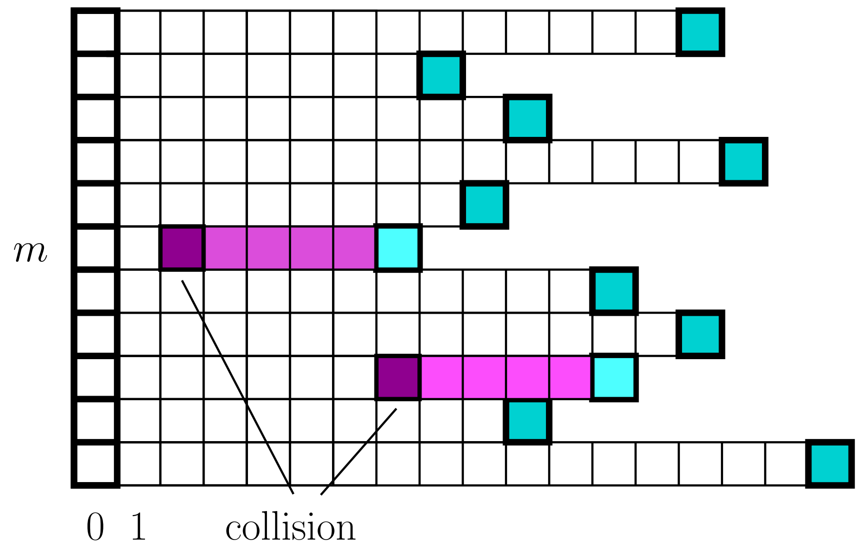
Attack 3 (new)
(Distinguished Points)

- Iterate until we reach a distinguished point



Attack 3 (new)
(Distinguished Points)

- Iterate until we reach a distinguished point



Attack 3 (new)
(Analysis)

- We assume that in total we need again about $\sqrt{n}$ data points
- Let $c = d/n$ be the ratio of distinguished points, $0 < c < 1$
- We assume that like for random points the average length of a row is about $1/c$

	Attack 1	Attack 2	Attack 3
Space complexity	$\sim \sqrt{\frac{n}{k}}$	$\sim \sqrt{n}/2$	$\sim c\sqrt{n}$
Data complexity	$\sim \sqrt{k\ n}$	$\sim \sqrt{n}$	$\sim \sqrt{n}$

Part 4

Conclusion

Conclusion

Entropy Estimator:

- We studied a stream cipher model with a random update function
- We introduced a new estimator of the state entropy after several iterations of the update function
- For small k it is more precise than the previous upper bound

Conclusion

Collision Attacks:

- Using a random update function introduces an entropy loss
- Till now it was not well studied if this introduce a real threat for our stream cipher model
- We showed that the proposed attacks are less effective than expected