Stream Ciphers Using a Random Update Function: Study of the Entropy of the Inner State

Andrea Röck
INRIA Paris-Rocquencourt, Team SECRET
France

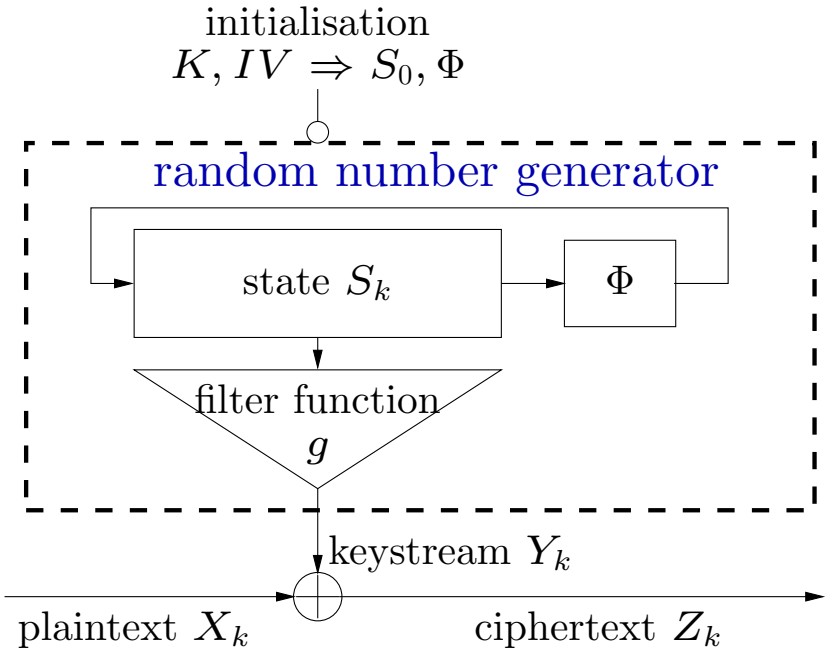
Africacrypt, June 12, 2008

Outline

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

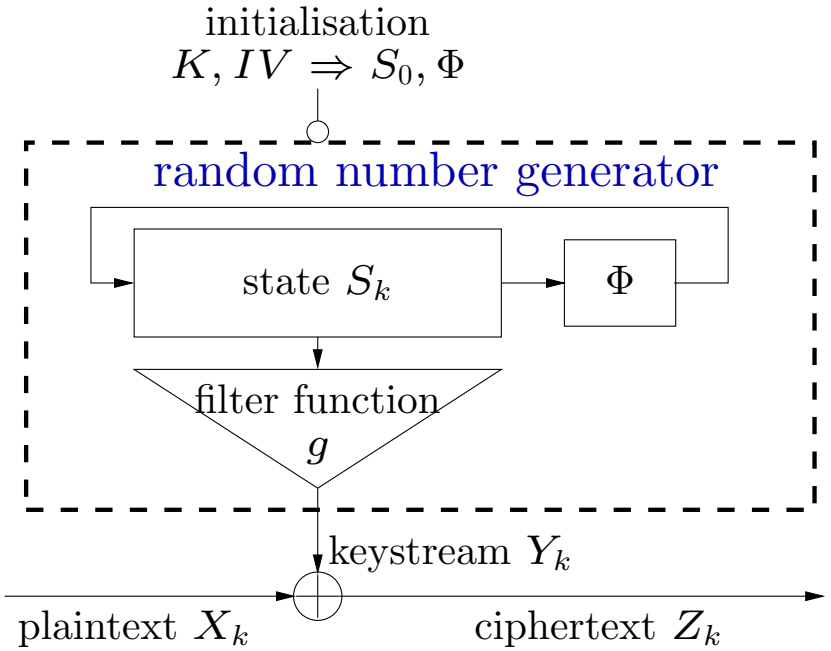
Part 1
Stream Cipher Model

Stream Cipher Model



- Initial State : S_0
- Update Function : $S_{k+1} = \Phi(S_k)$ for $k \geq 0$
- Keystream : $Y_k = g(S_k)$
- Ciphertext : $Z_k = X_k \oplus Y_k$

Probabilistic Model (Information of an adversary)



- State space : $\Omega_n = \{\omega_1, \omega_2, \ldots \omega_n\}$
- Initial distribution : $\{p_i\}_{i=1}^n$ with $p_i = Pr[S_0 = \omega_i]$
- *Random update function* : $Pr[\Phi = \varphi] = 1/n^n$
  for all $\varphi \in \mathcal{F}_n = \{\varphi : \Omega_n \to \Omega_n\}$

State Entropy

- **Probability** that the state has the value ω_i after k iterations of Φ :

$$p_i^{\Phi}(k) = Pr[S_k = \omega_i] = Pr[\Phi^k(S_0) = \omega_i]$$

- **Shannon's entropy H :** is a measure of the information contained in a random variable. It must hold that :

$$H \leq \log_2(n)$$

- **State entropy** after k iterations of Φ :

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

- **Average state entropy** after k iterations, taken over all functions $\varphi \in \mathcal{F}_n$:

$$\mathbf{H_k} = \mathbf{E}(H_k^{\Phi})$$

Motivation

- A *random function* allows us to study some interesting properties of our stream cipher model, on average over all function $\varphi \in \mathcal{F}_n$.
- Some new stream ciphers use an update function which behaves almost like a random function. (*e.g.* : the eSTREAM candidate MICKEY (version 1) **[Babbage et Dodd 05]**)
- The image size of Φ^k is smaller than n, thus we *loose entropy*.

Questions :

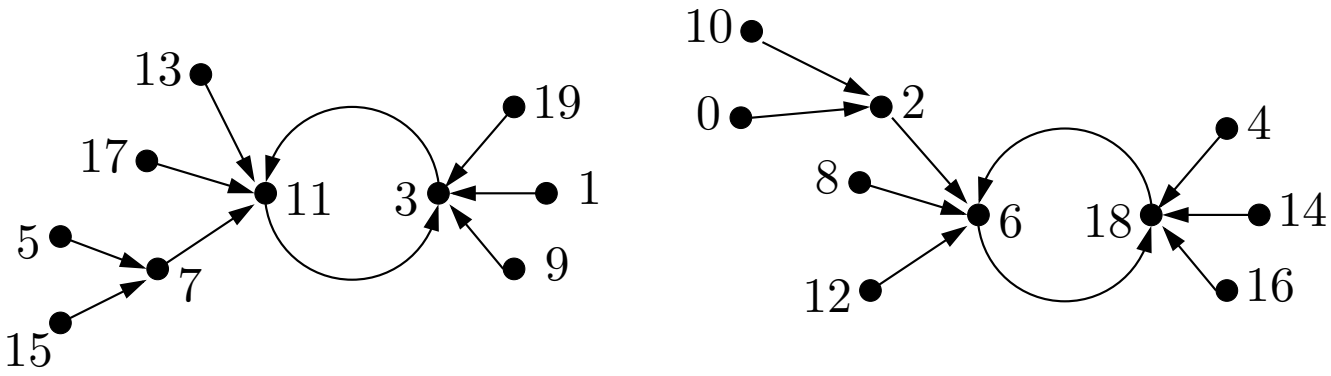
- How much entropy do we loose in the internal state?
- Can this loss be efficiently exploited into a collision attack?

Part 2

Entropy Estimation

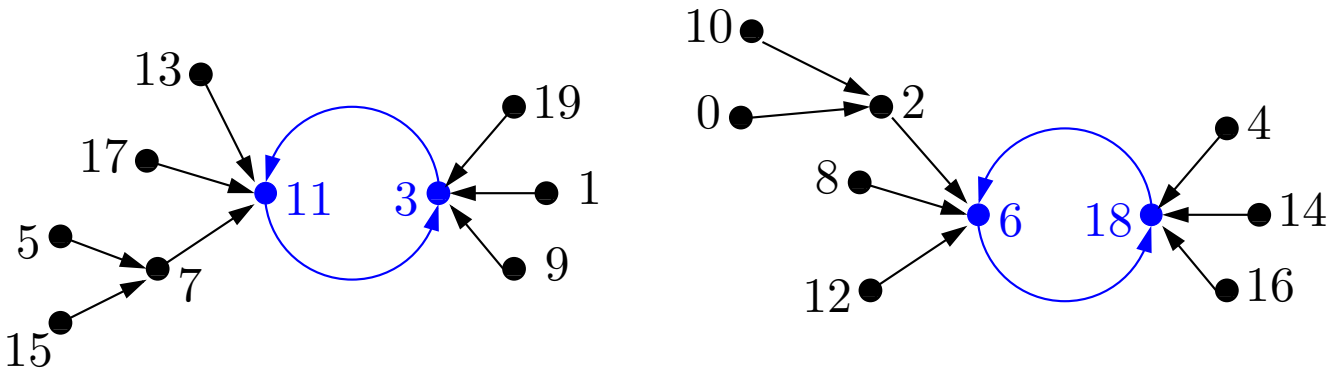
Previous Results

Example of a Functional Graph



$$\varphi : x \to x^2 + 2 \pmod{20}$$

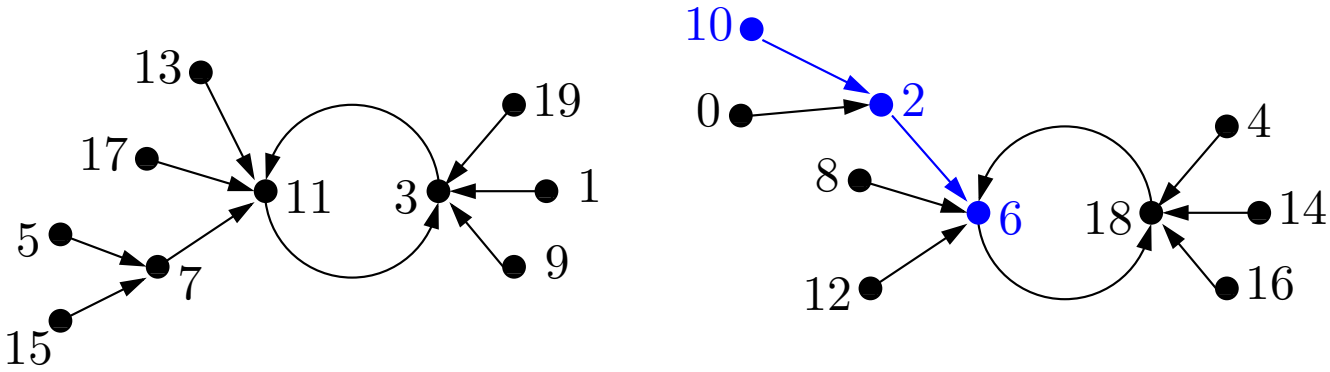
Properties of Random Functions [Flajolet Odlyzko 90]



Asymptotic values for $n \to \infty$:

- Expected number of cycle points : $\mathbf{cp}(n) \sim \sqrt{\pi n/2}$

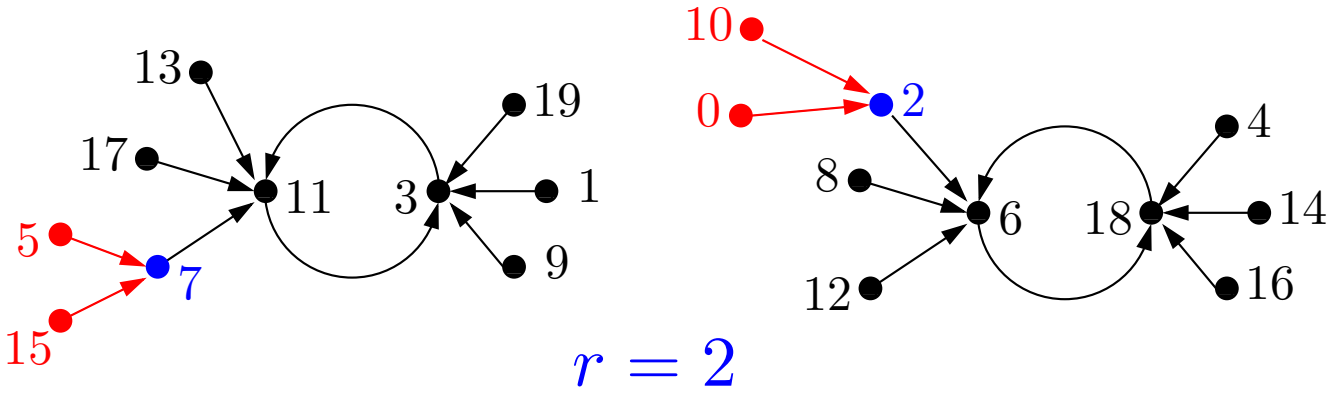
Properties of Random Functions [Flajolet Odlyzko 90]



Asymptotic values for $n \to \infty$:

- Expected number of cycle points : $\mathbf{cp}(n) \sim \sqrt{\pi n/2}$
- Expected maximal tail length : $\mathbf{mt}(n) \sim \sqrt{\pi n/8}$

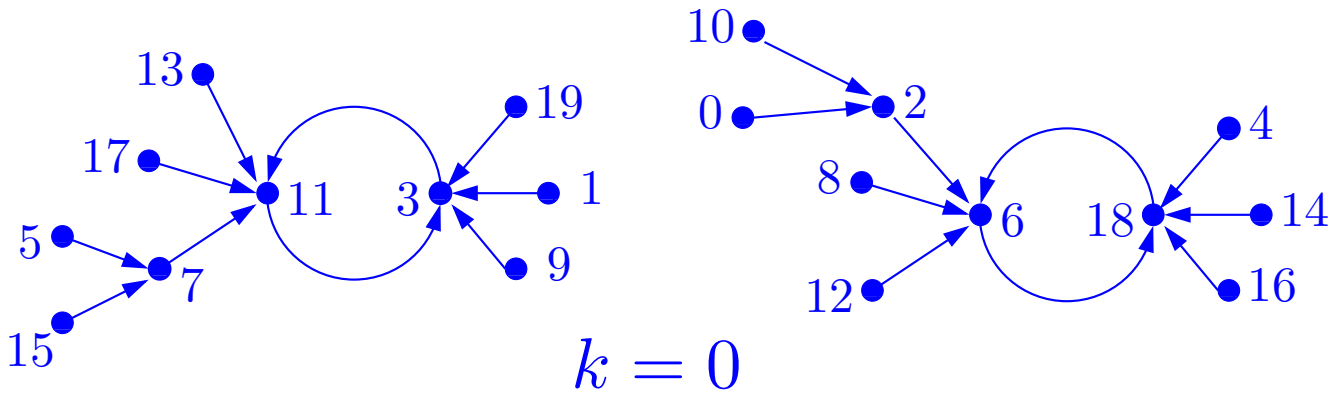
Properties of Random Functions [Flajolet Odlyzko 90]



Asymptotic values for $n \to \infty$:

- Expected number of cycle points : $\mathbf{cp}(n) \sim \sqrt{\pi n/2}$
- Expected maximal tail length : $\mathbf{mt}(n) \sim \sqrt{\pi n/8}$
- Expected number of r-nodes : $\mathbf{rn}(n, r) \sim \frac{n}{r!e}$

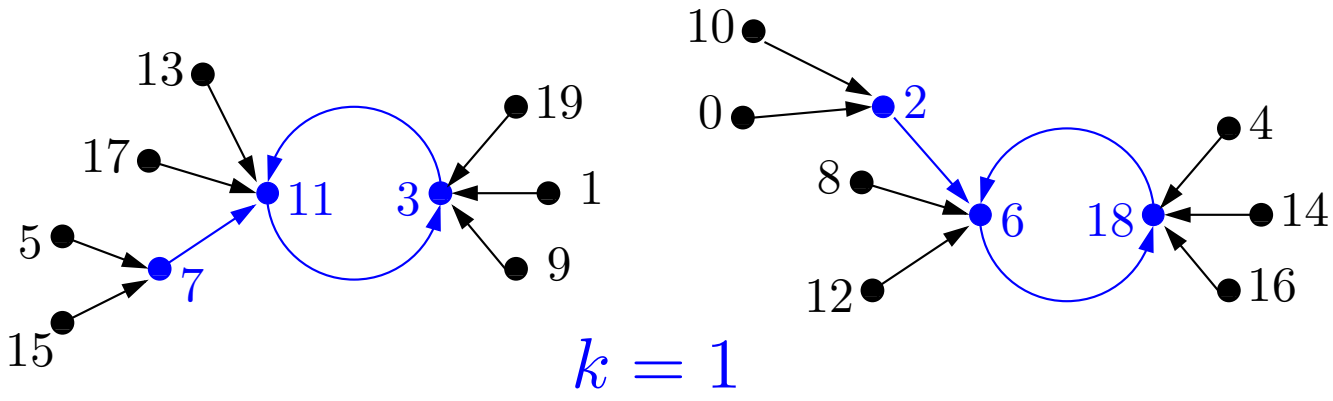
Properties of Random Functions [Flajolet Odlyzko 90]



Asymptotic values for $n \to \infty$:

- Expected number of cycle points : $\mathbf{cp}(n) \sim \sqrt{\pi n/2}$
- Expected maximal tail length : $\mathbf{mt}(n) \sim \sqrt{\pi n/8}$
- Expected number of r-nodes : $\mathbf{rn}(n, r) \sim \frac{n}{r!e}$
- Expected number of image points : $\mathbf{ip}(n, k) \sim n(1 - \tau_k)$
  where $\tau_0 = 0$ and $\tau_{k+1} = e^{-1+\tau_k}$

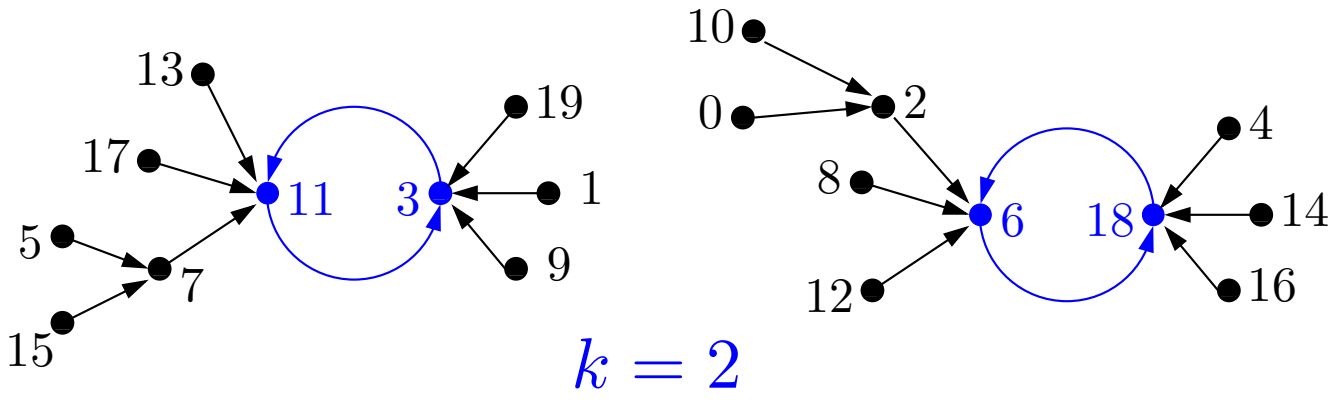
Properties of Random Functions [Flajolet Odlyzko 90]



Asymptotic values for $n \to \infty$:

- Expected number of cycle points : $\mathbf{cp}(n) \sim \sqrt{\pi n/2}$
- Expected maximal tail length : $\mathbf{mt}(n) \sim \sqrt{\pi n/8}$
- Expected number of r-nodes : $\mathbf{rn}(n, r) \sim \frac{n}{r!e}$
- Expected number of image points : $\mathbf{ip}(n, k) \sim n(1 - \tau_k)$
  where $\tau_0 = 0$ and $\tau_{k+1} = e^{-1+\tau_k}$

Properties of Random Functions [Flajolet Odlyzko 90]



$k = 2$

Asymptotic values for $n \to \infty$:

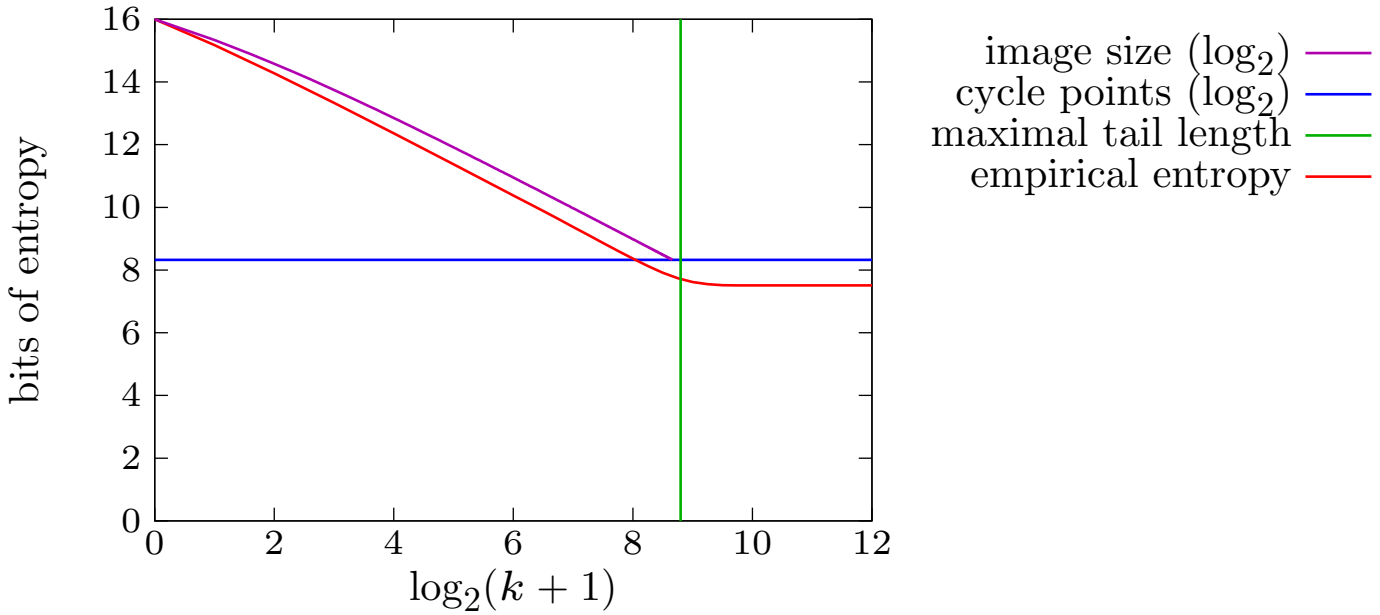
- Expected number of cycle points : $\mathbf{cp}(n) \sim \sqrt{\pi n/2}$
- Expected maximal tail length : $\mathbf{mt}(n) \sim \sqrt{\pi n/8}$
- Expected number of r-nodes : $\mathbf{rn}(n, r) \sim \frac{n}{r!e}$
- Expected number of image points : $\mathbf{ip}(n, k) \sim n(1 - \tau_k)$
  where $\tau_0 = 0$ and $\tau_{k+1} = e^{-1+\tau_k}$

Bounding Entropy with Image Points

- Upper bound given by number of image points **[Hong Kim 05]** :

$$\mathbf{H_k} \leq log_2(n) + log_2(1 - \tau_k)$$

- 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)

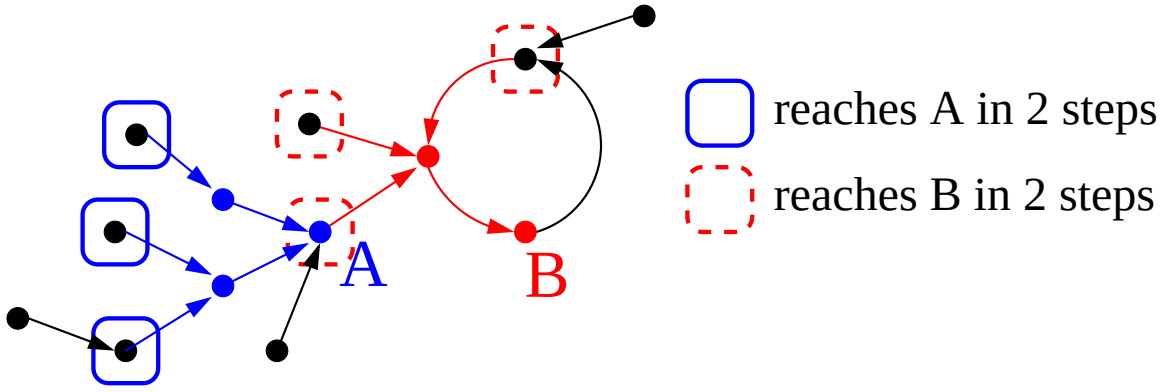
- **Definition :**

  Number of points which are reached by r points after k iterations.

$$\varphi \in \mathcal{F}_n : \quad \mathrm{rn}_k^{\varphi}(r) = \#\{i |\ |\varphi^{-k}(i)| = r\}$$

$$\text{Average} : \quad \mathbf{rn_k}(n,r) = \frac{1}{n^n} \sum_{\varphi \in \mathcal{F}_n} \mathrm{rn}_k^{\varphi}(r)$$

- **Example for $k = 2$ and $r = 3$:**



New Entropy Estimator (3)

- **Theorem :**
  For a uniform initial distribution the expected entropy of the inner state after k iterations is :

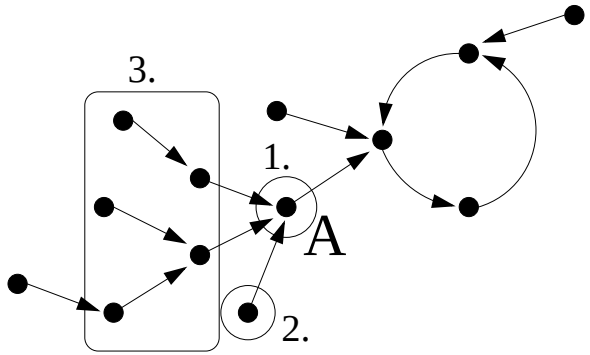
$$\mathbf{H_k} = \log_2(n) - \sum_{r=1}^{n} \mathbf{rn_k}(n,r)\ \frac{r}{n}\ \log_2(r)$$

- **Theorem :**
  For an arbitrary initial distribution $P = \{p_1, p_2, \ldots, p_n\}$ the expected entropy of the inner state after k iterations is :

$$\mathbf{H_k^P} = \sum_{r=1}^{n} \mathbf{rn_k}(n,r) \frac{1}{\binom{n}{r}} \sum_{1 \le j_1 < \cdots < j_r \le n} (p_{j_1} + \cdots + p_{j_r}) \log_2 \frac{1}{p_{j_1} + \cdots + p_{j_r}}$$

Computation of $\mathbf{rn_k}(n, r)$ (1)

- $k = 1$: Use directly the asymptotic value $\mathbf{rn}(n, r) \sim \frac{n}{r!e}$
- $k > 1$: Use the fact that such a tree node consists of :
  1. A *node*.
  2. A *SET* of *trees* with a depth $< k - 1$.
  3. A *CONCATENATION* of j *trees* of depth $\geq k - 1$ and $1 \leq j \leq r$. Their roots are reached by respectively $i_1, \ldots, i_j$ nodes after $k - 1$ iterations such that $i_1 + \cdots + i_j = r$.



Computation of $\mathbf{rn_k}(n, r)$ (2)

- By analyzing the generating function of our property we find a $c_k(r)$ such that for $n \longrightarrow \infty$:

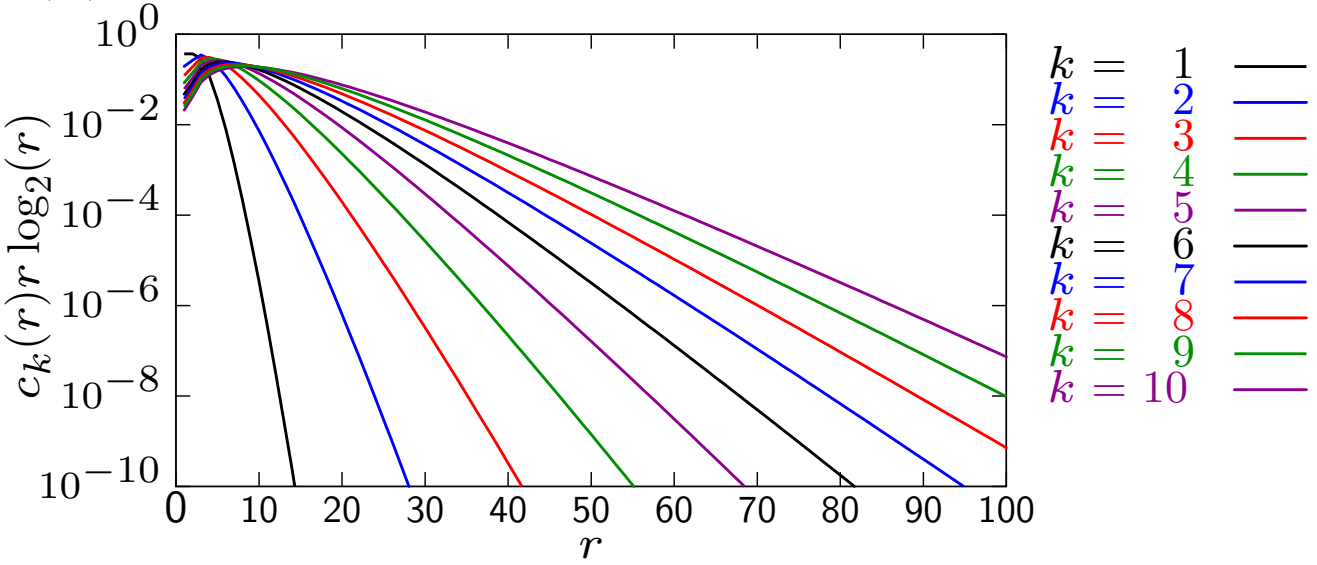
$$\mathbf{rn_k}(n, r) \sim n\ c_k(r)$$

- We can compute $c_k(r)$ in $O\left(k\ r^2 \log(r)\right)$.
- For a **uniform initial distribution** we can write :

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

Remarks

- $c_k(r) r \log_2(r)$ decreases very fast.



- **Approximation :**

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

- We ignore the incoming cycle nodes.

Estimation of Entropy Loss with different Methods

k		1	2	3	10	50	100
empirical data $n = 2^{16}$		0.8273	1.3458	1.7254	3.1130	5.2937	6.2529
image points		0.6617	1.0938	1.4186	2.6599	4.7312	5.6913
$H_k(R)$	$R = 50$	0.8272	1.3457	1.7254	3.1084	2.6894	1.2524
	$R = 200$	0.8272	1.3457	1.7254	3.1129	5.2661	5.5172
	$R = 1000$	0.8272	1.3457	1.7254	3.1129	5.2918	6.2729

- 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 4

Collision Attacks

Collision Attacks (1)

- **Collision** :
  - Different initial states S_0, S'_0 and $k, k' >= 0$ such that $S_k = S'_{k'}$.
  - A given S_0, $k, k' >= 0$ and $k \neq k'$ such that $S_k = S_{k'}$.
- We compare the attack with a direct search for a collision in the initial state.
- **Three criteria** :
  - Number of initial states.
  - Space complexity.
  - Query complexity.

Collision Attacks (2)

Ideas:

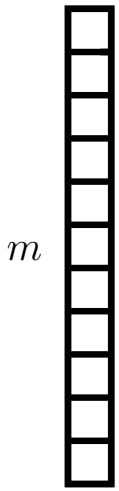
- Using a random function leads to a loss of entropy.
- A reduced entropy leads to higher probability of a collision.
- If two states are the same, then the subsequent output sequences are identical.
- Two proposals for an attack on MICKEY in **[Hong Kim 05]** (no real attacks).

Attack 1
(Proposition [Hong Kim 05])

- Search for collision after k iterations.

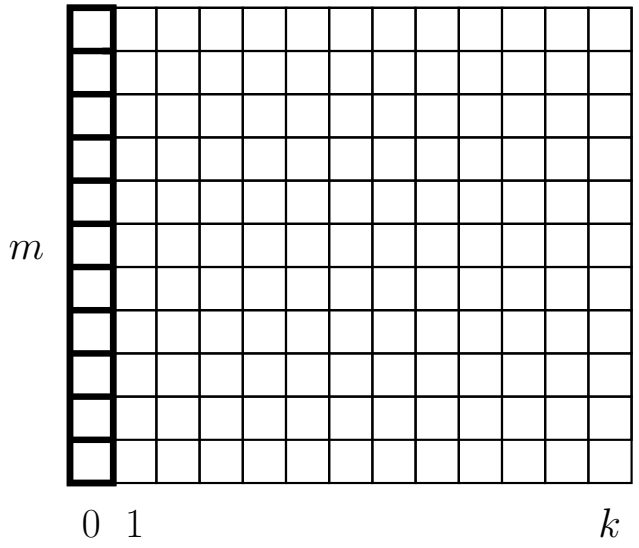
Attack 1
(Proposition [Hong Kim 05])

- Search for collision after k iterations.



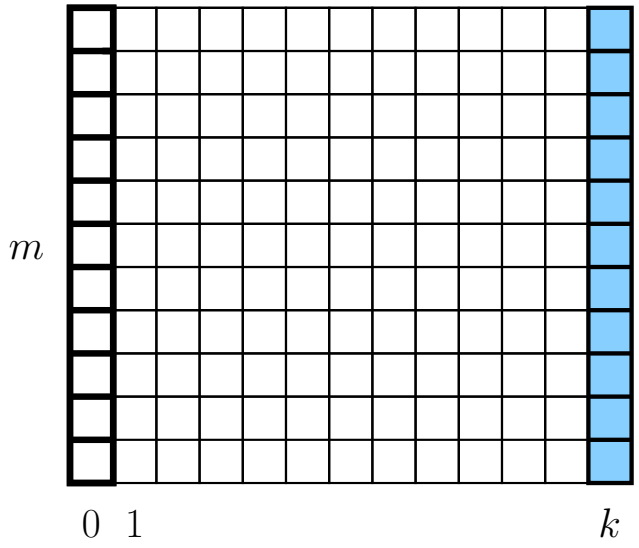
Attack 1
(Proposition [Hong Kim 05])

- Search for collision after k iterations.



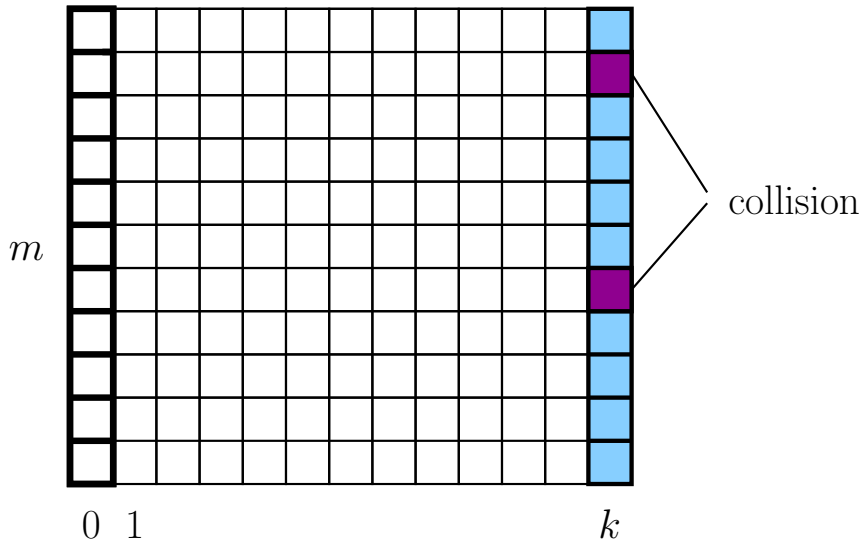
Attack 1
(Proposition [Hong Kim 05])

- Search for collision after k iterations.



Attack 1
(Proposition [Hong Kim 05])

- Search for collision after k iterations.



Attack 1
(Analysis)

- **Upper bound:** $\mathbf{H_k} \leq \log_2(n) - \log_2(k) + 1$
- **Birthday paradox:** Need $\sim \sqrt{2n/k}$ values in the last row.

	Attack 1
# initial states	$\sim \sqrt{\frac{2n}{k}}$
space complexity **[Hong Kim 05]**	$\sim \sqrt{\frac{2n}{k}}$
query complexity **(new)**	$\sim \sqrt{2kn}$

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.

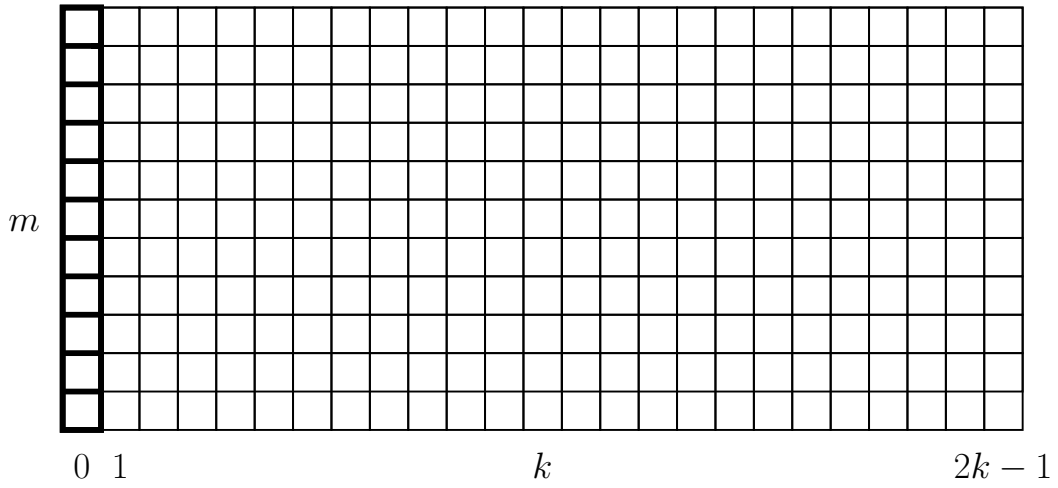
This means that we don't use a random function, but the principle of the attack stays the same.

Attack 2
(Proposition [Hong Kim 05])

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

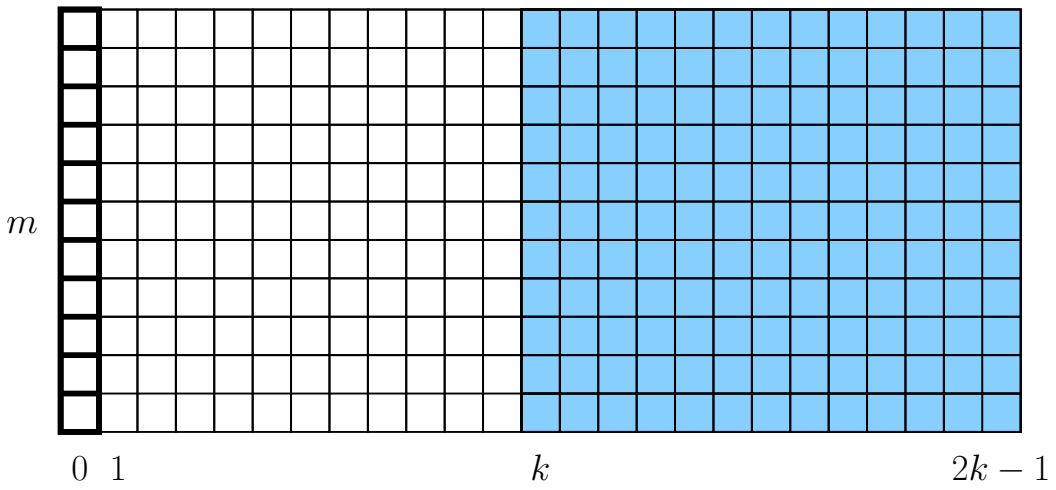
Attack 2
(Proposition [Hong Kim 05])

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



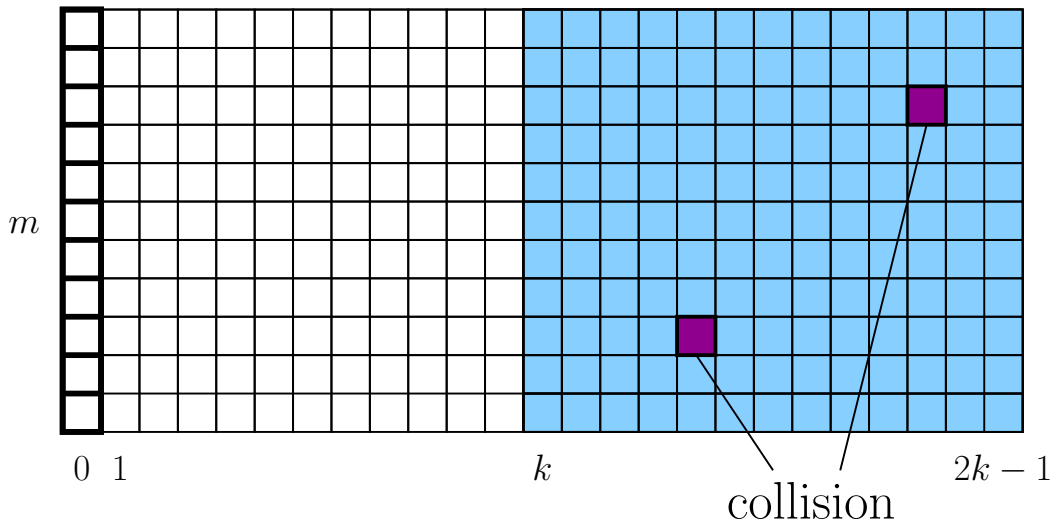
Attack 2
(Proposition [Hong Kim 05])

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



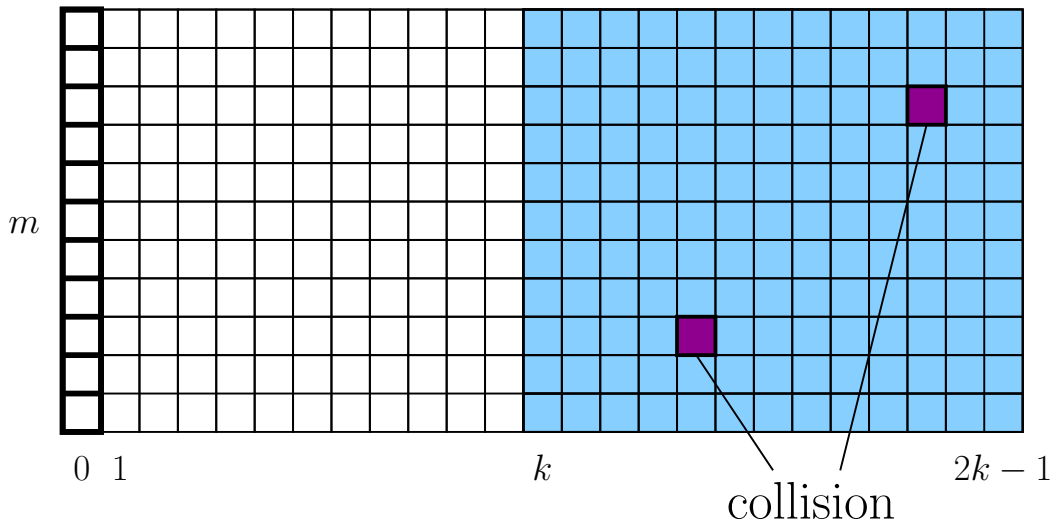
Attack 2
(Proposition [Hong Kim 05])

- 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))

- Let $Pr[A]$ be the probability of no collision in the $2km$ points.
- Probability of collision in km points is smaller than $1 - Pr[A]$.
- By counting arguments : $Pr[A] = \frac{n(n-1)\cdots(n-2km+1)}{n^{2km}}$
- **Birthday Paradox:** We need $2mk \approx \sqrt{n}$

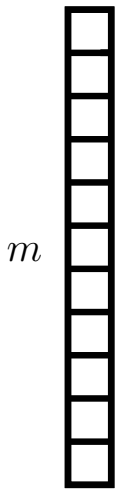
	Attack 1	Attack 2
# initial states	$\sim \sqrt{\frac{2n}{k}}$	$\sim \frac{\sqrt{n}}{2k}$
space complexity	$\sim \sqrt{\frac{2n}{k}}$	$\sim \frac{\sqrt{n}}{2}$
query complexity	$\sim \sqrt{2kn}$	$\sim \sqrt{n}$

Attack 3 (new)
(Distinguished Points)

- Iterate until we reach a distinguished point.

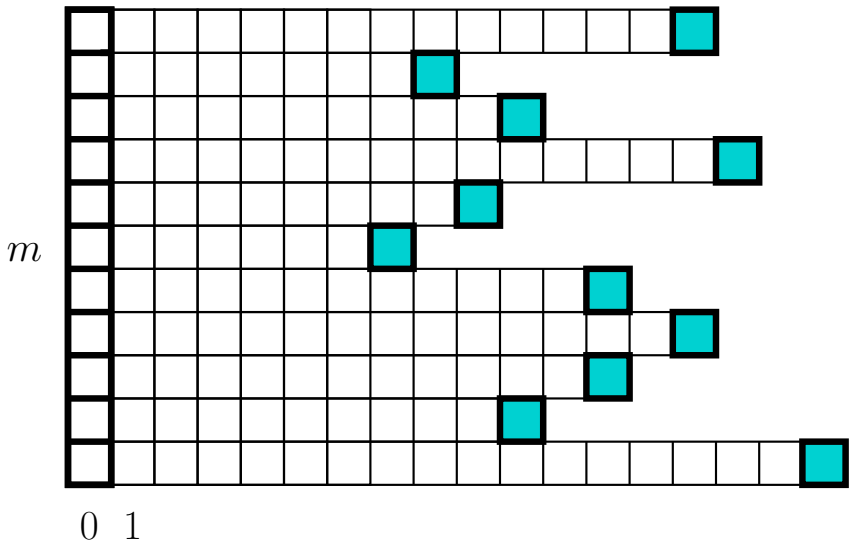
Attack 3 (new)
(Distinguished Points)

- Iterate until we reach a distinguished point.



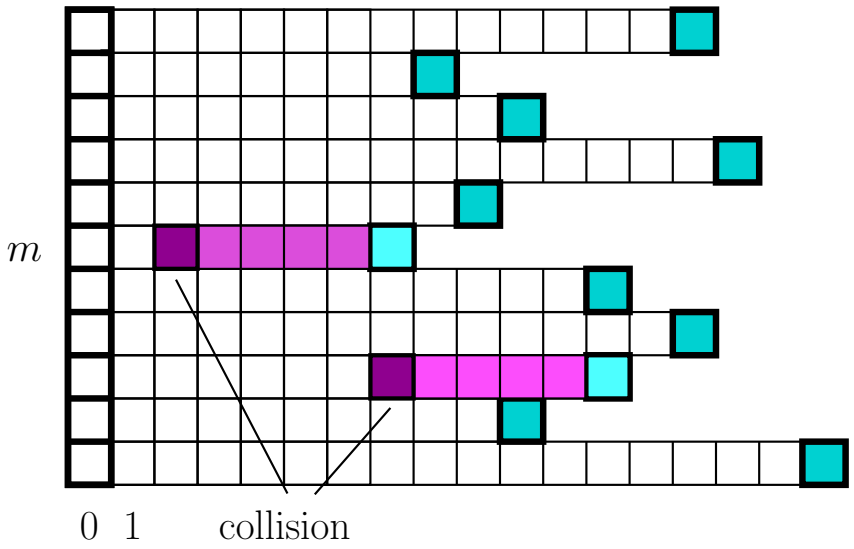
Attack 3 (new)
(Distinguished Points)

- 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 $1/c$.
- *E.g.* $n = 2^{20}$, $k_{max} = \sqrt{n}$, and $0.7 \leq \frac{\log_2(d)}{\log_2(n)} \leq 1$ (*i.e.* $2^{-6} \leq c \leq 1$).

	Attack 1	Attack 2	Attack 3
# initial states	$\sim \sqrt{\frac{2n}{k}}$	$\sim \frac{\sqrt{n}}{2k}$	$\sim c\sqrt{n}$
space complexity	$\sim \sqrt{\frac{2n}{k}}$	$\sim \frac{\sqrt{n}}{2}$	$\sim c\sqrt{n}$
query complexity	$\sim \sqrt{2kn}$	$\sim \sqrt{n}$	$\sim \sqrt{n}$

Part 5

Conclusion

Conclusion

Entropy Estimator:

- We studied a stream cipher model with a random update function.
- We introduced a new estimator which can be iteratively computed.
- 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.