Analysis and Implementation of RC4 Stream Cipher

Sourav Sen Gupta

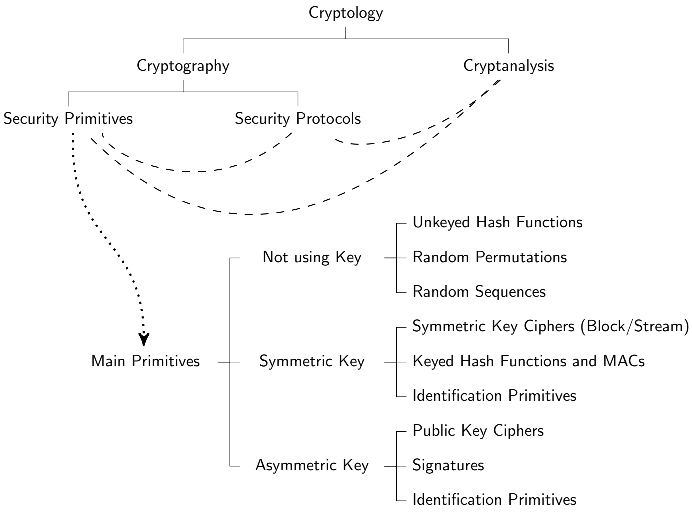
A thesis presented to Indian Statistical Institute in fulfillment of the thesis requirement for the degree of Doctor of Philosophy in Computer Science.

Under the supervision of
Professor Subhamoy Maitra
Applied Statistics Unit, ISI Kolkata

ISI Kolkata

6 January 2014

Scope of the Thesis



Scope of the Thesis

- Cryptology
  - Cryptography
    - Security Primitives
    - Security Protocols
  - Cryptanalysis

Stream Cipher **RC4**

- Main Primitives
  - Not using Key
    - Unkeyed Hash Functions
    - Random Permutations
    - Random Sequences
  - Symmetric Key
    - Symmetric Key Ciphers (Block/Stream)
    - Keyed Hash Functions and MACs
    - Identification Primitives
  - Asymmetric Key
    - Public Key Ciphers
    - Signatures
    - Identification Primitives

Organization of the Thesis

There are 9 chapters, distributed over 2 major parts, in this thesis.

Chapter 1 – Preliminaries and Motivation	
Part I – Analysis of RC4	Part II – Implementation of RC4
Chapter 2 – Overview of RC4 Analysis	Chapter 6 – Overview of RC4 Designs
Chapter 3 – Keylength biases	Chapter 7 – Design 1 (loop unrolling)
Chapter 4 – State-dependent biases	Chapter 8 – Design 2 (pipelining)
Chapter 5 – Keystream biases	
Chapter 9 – Conclusion and Open Problems	

We deal with 10 research problems in this thesis.
We present 10 open problems in related research.

Organization of this Talk

Stream Ciphers and RC4

Stream Ciphers

Exploit the notion of *perfect secrecy* by Shannon, 1949.

Encrypted message reveals no information about the plaintext for a *one-time-pad* encryption.

random keystream

$\bigoplus$

plaintext message

---

encrypted message

Shannon, Claude E. (October 1949). "Communication Theory of Secrecy Systems". Bell System Technical Journal (USA: AT&T Corporation) 28 (4):656–715.

Stream Ciphers

Exploit the notion of *perfect secrecy* by Shannon, 1949.

$$\text{secret key} \longrightarrow \boxed{\text{SC}} \longrightarrow \text{random keystream}$$

$$\begin{array}{c} \text{random keystream} \\ \oplus \\ \text{plaintext message} \\ \hline \text{encrypted message} \end{array}$$

They aim at producing *long* random keystream from a *short* secret key.

Shannon, Claude E. (October 1949). "Communication Theory of Secrecy Systems". Bell System Technical Journal (USA: AT&T Corporation) 28 (4):656–715.

Stream Ciphers

Exploit the notion of *perfect secrecy* by Shannon, 1949.

secret key $\longrightarrow$ SC $\longrightarrow$ pseudo random keystream

$\oplus$

plaintext message

---

encrypted message

They aim at producing *long* random keystream from a *short* secret key.

But never produce a truly random keystream!

Shannon, Claude E. (October 1949). "Communication Theory of Secrecy Systems". Bell System Technical Journal (USA: AT&T Corporation) 28 (4):656–715.

RC4 Stream Cipher

- Designed by Ron Rivest in 1987
- Description public in 1994 (?)

Photo: http://people.csail.mit.edu/rivest/

Popularity

- Most used commercial cipher!
- Used in WEP, WPA, SSL/TLS.
- Numerous academic publications and patents.

RC4 Stream Cipher

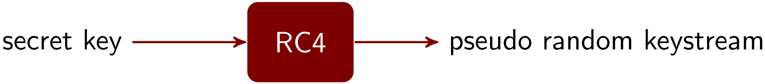
- Designed by Ron Rivest in 1987
- Description public in 1994 (?)

Photo: http://people.csail.mit.edu/rivest/

Popularity

- Most used commercial cipher!
- Used in WEP, WPA, SSL/TLS.
- Numerous academic publications and patents.
- Simplest cipher to describe!

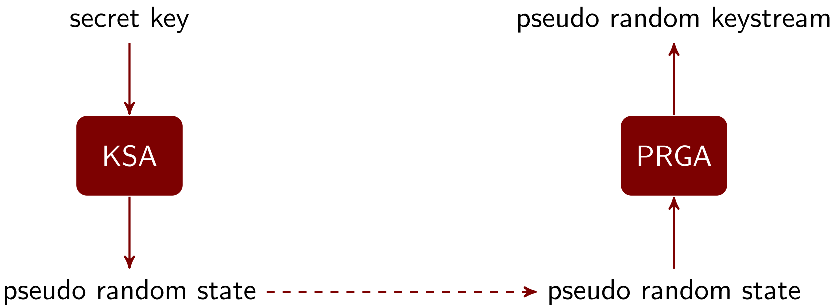
RC4 Stream Cipher



RC4 Stream Cipher

Key Scheduling Algorithm

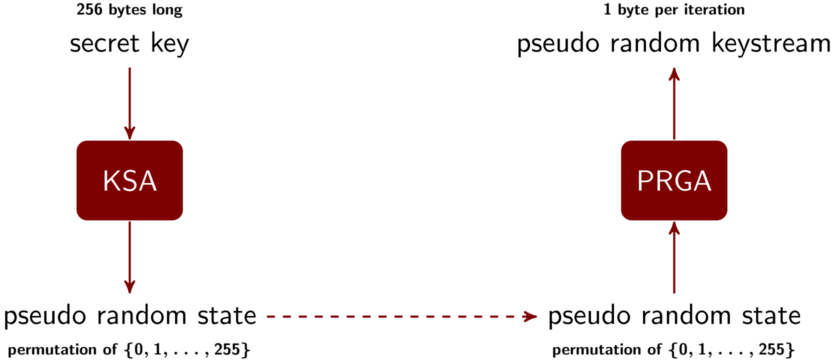
Pseudo-Random Generation Algorithm



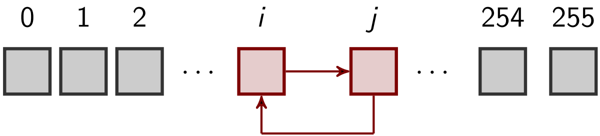
RC4 Stream Cipher

Key Scheduling Algorithm

Pseudo-Random Generation Algorithm



Key Scheduling Algorithm (KSA)



Initialize index: $j = 0$;

for $i = 0, \ldots, 255$ **do**

  $j = j + S[i] + K[i]$;

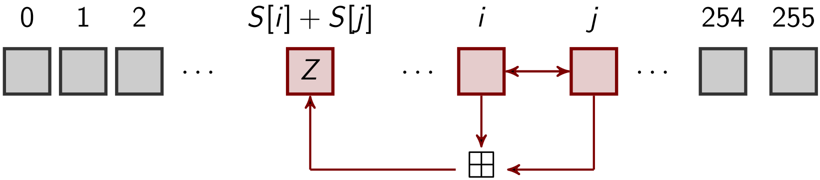
  Swap $S[i] \leftrightarrow S[j]$;

end

INPUT: S-array initialized to identity permutation, and key K

OUTPUT: Scrambled S-array

Pseudo-Random Generation Algorithm (PRGA)



Initialize indices: $i = j = 0$;

while *TRUE* **do**
  $i = i + 1$;
  $j = j + S[i]$;
  Swap $S[i] \leftrightarrow S[j]$;
  Output $Z = S[S[i] + S[j]]$;
end

INPUT: Scrambled S-array, obtained as the KSA output

OUTPUT: Pseudo-random stream

RC4 toy example

KSA with $N = 8$

```
K = [3, 1, 5, 2, 7, 0, 6, 4]
S = [0, 1, 2, 3, 4, 5, 6, 7]

i = 0 , j = 3       S = [3, 1, 2, 0, 4, 5, 6, 7]
i = 1 , j = 5       S = [3, 5, 2, 0, 4, 1, 6, 7]
i = 2 , j = 4       S = [3, 5, 4, 0, 2, 1, 6, 7]
i = 3 , j = 6       S = [3, 5, 4, 6, 2, 1, 0, 7]
i = 4 , j = 7       S = [3, 5, 4, 6, 7, 1, 0, 2]
i = 5 , j = 0       S = [1, 5, 4, 6, 7, 3, 0, 2]
i = 6 , j = 6       S = [1, 5, 4, 6, 7, 3, 0, 2]
i = 7 , j = 4       S = [1, 5, 4, 6, 2, 3, 0, 7]
```

RC4 toy example

PRGA with $N = 8$

```
K is no more required
S = [1, 5, 4, 6, 2, 3, 0, 7]

i = 1 , j = 5      S = [1, 3, 4, 6, 2, 5, 0, 7] , Z = 1
i = 2 , j = 1      S = [1, 4, 3, 6, 2, 5, 0, 7] , Z = 7
i = 3 , j = 7      S = [1, 4, 3, 7, 2, 5, 0, 6] , Z = 5
i = 4 , j = 1      S = [1, 2, 3, 7, 4, 5, 0, 6] , Z = 0
i = 5 , j = 6      S = [1, 2, 3, 7, 4, 0, 5, 6] , Z = 0
i = 6 , j = 3      S = [1, 2, 3, 5, 4, 0, 7, 6] , Z = 4
i = 7 , j = 1      S = [1, 6, 3, 5, 4, 0, 7, 2] , Z = 1
i = 8 , j = ...    S = ...
```

How can a design so simple have

such enigmatic a flair?!

How can a design so simple have

such enigmatic a flair?!

Used in three main protocols WEP, WPA, SSL/TLS
Numerous applications in Microsoft, Apple, SQL products
Prominent patents on hardware implementation

How can a design so simple have

such enigmatic a flair?!

Used in three main protocols WEP, WPA, SSL/TLS
Numerous applications in Microsoft, Apple, SQL products
Prominent patents on hardware implementation

More than hundred papers in top-tier venues
Three Master's theses, two PhD theses, one Book to date

How can a design so simple have

such enigmatic a flair?!

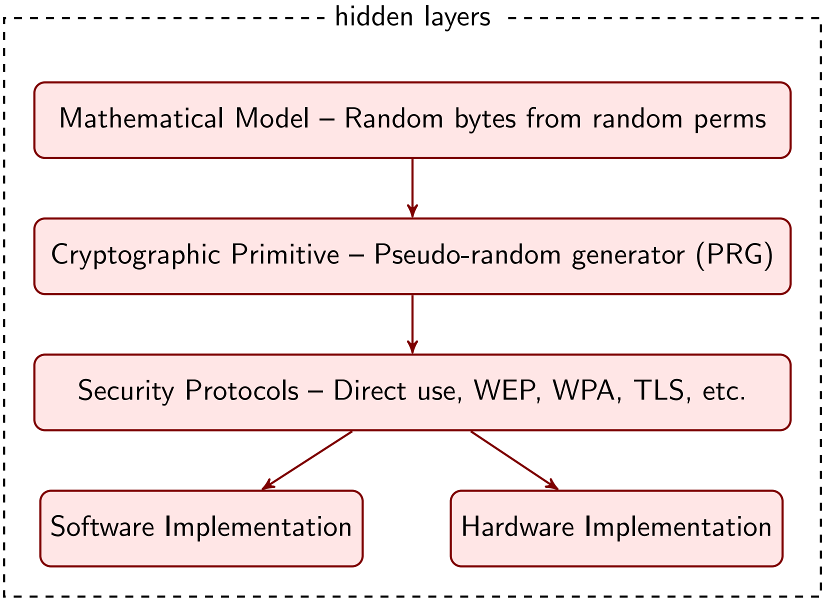
Used in three main protocols WEP, WPA, SSL/TLS
Numerous applications in Microsoft, Apple, SQL products
Prominent patents on hardware implementation

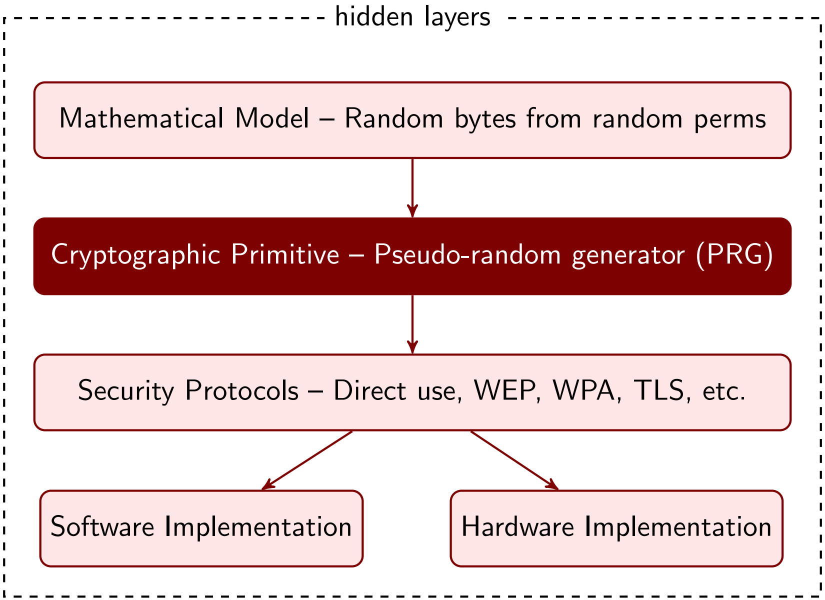
More than hundred papers in top-tier venues
Three Master's theses, two PhD theses, one Book to date

One must *cultivate* this cipher!

Part I

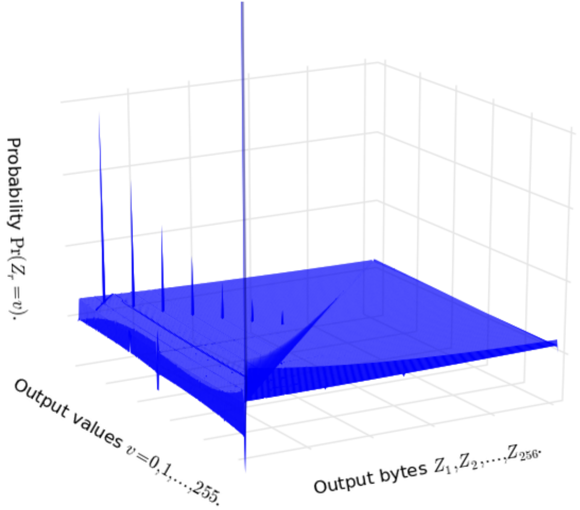
Analysis of RC4


hidden layers
Mathematical Model – Random bytes from random perms
Cryptographic Primitive – Pseudo-random generator (PRG)
Security Protocols – Direct use, WEP, WPA, TLS, etc.
Software Implementation
Hardware Implementation


hidden layers
Mathematical Model – Random bytes from random perms
Cryptographic Primitive – Pseudo-random generator (PRG)
Security Protocols – Direct use, WEP, WPA, TLS, etc.
Software Implementation
Hardware Implementation

PRG output should be *indistinguishable* from truly random bitstream!

PRG output should be *indistinguishable* from truly random bitstream!



Broadcast attack on RC4

Encryption using RC4 is typically $E(k, P) : C \leftarrow P \oplus RC4(k)$

$$C_1 = P_1 \oplus Z_1, \quad C_2 = P_2 \oplus Z_2, \quad C_3 = P_3 \oplus Z_3, \quad \ldots$$

Broadcast attack on RC4

Encryption using RC4 is typically $E(k, P) : C \leftarrow P \oplus RC4(k)$

$$C_1 = P_1 \oplus Z_1, \quad C_2 = P_2 \oplus Z_2, \quad C_3 = P_3 \oplus Z_3, \quad \ldots$$

Mantin-Shamir (2001): $\Pr(Z_2 = 0) \approx 2/N$

Broadcast attack on RC4

Encryption using RC4 is typically $E(k,P) : C \leftarrow P \oplus RC4(k)$

$$C_1 = P_1 \oplus Z_1, \quad C_2 = P_2 \oplus Z_2, \quad C_3 = P_3 \oplus Z_3, \quad \ldots$$

Mantin-Shamir (2001): $\Pr(Z_2 = 0) \approx 2/N \;\Rightarrow\; \Pr(C_2 = P_2) \approx 2/N$

Broadcast attack on RC4

Encryption using RC4 is typically $E(k, P) : C \leftarrow P \oplus RC4(k)$

$$C_1 = P_1 \oplus Z_1, \quad C_2 = P_2 \oplus Z_2, \quad C_3 = P_3 \oplus Z_3, \quad \ldots$$

Mantin-Shamir (2001): $\Pr(Z_2 = 0) \approx 2/N \;\Rightarrow\; \Pr(C_2 = P_2) \approx 2/N$

Consider a ciphertext-only-attack where the same plaintext P is encrypted by RC4 several times using independent random keys.

Broadcast attack on RC4

Encryption using RC4 is typically $E(k, P) : C \leftarrow P \oplus RC4(k)$

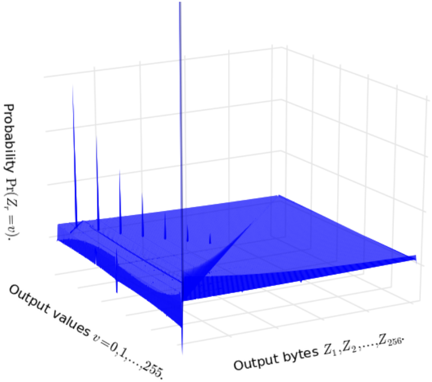
$$C_1 = P_1 \oplus Z_1, \quad C_2 = P_2 \oplus Z_2, \quad C_3 = P_3 \oplus Z_3, \quad \dots$$

Mantin-Shamir (2001): $\Pr(Z_2 = 0) \approx 2/N \;\Rightarrow\; \Pr(C_2 = P_2) \approx 2/N$

Consider a ciphertext-only-attack where the same plaintext P is encrypted by RC4 several times using independent random keys.

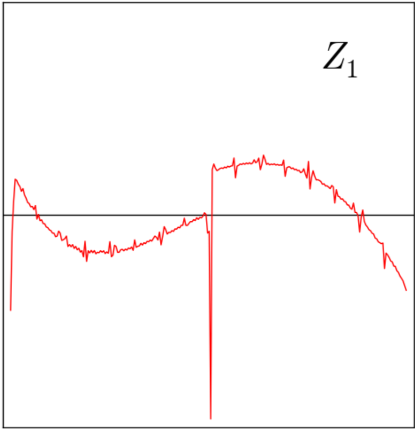
Plaintext recovery

- Gather multiple C and compute $P_2 = \text{majority}\{C_2\}$
- Attack will be successful if number of C is in $\Omega(N)$



Non-randomness in initial bytes

Non-randomness in Z_1



$\Pr(Z_1 = v)$
$v = 0, 1, \ldots, 255$

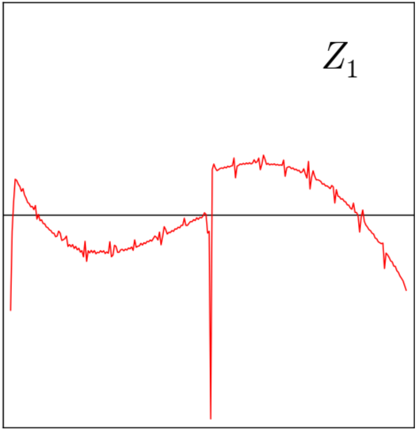
Major biases

Sinusoidal distribution

$\Pr(Z_1 = 0) \approx \frac{1}{N} - \frac{1}{N^2}$

$\Pr(Z_1 = 129) \approx \frac{1}{N} - \frac{2}{N^2}$

Non-randomness in Z_1



$\Pr(Z_1 = v)$
$v = 0, 1, \ldots, 255$

Major biases

Sinusoidal distribution

$\Pr(Z_1 = 0) \approx \frac{1}{N} - \frac{1}{N^2}$

$\Pr(Z_1 = 129) \approx \frac{1}{N} - \frac{2}{N^2}$

Mironov, Crypto 2002

Negative bias in $(Z_1 = 0)$

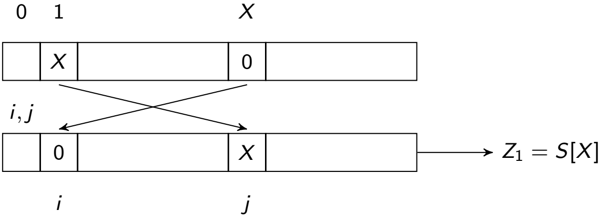
Theorem

Suppose the initial permutation of RC4 PRGA is a random permutation of $\{0, 1, \ldots, N-1\}$*. Then* $\Pr(Z_1 = 0) \approx 1/N - 1/N^2$.

Negative bias in $(Z_1 = 0)$

Theorem

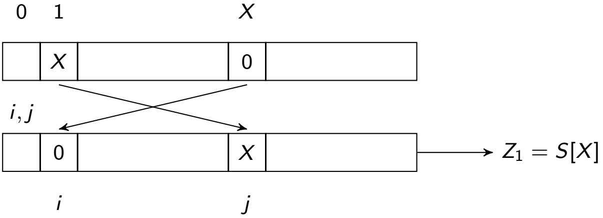
Suppose the initial permutation of RC4 PRGA is a random permutation of $\{0, 1, \ldots, N-1\}$. *Then* $\Pr(Z_1 = 0) \approx 1/N - 1/N^2$.



Negative bias in $(Z_1 = 0)$

Theorem

Suppose the initial permutation of RC4 PRGA is a random permutation of $\{0, 1, \ldots, N-1\}$. Then $\Pr(Z_1 = 0) \approx 1/N - 1/N^2$.



$$\Pr(Z_1 = 0) \approx 0 \cdot 1/N + 1/N \cdot (1 - 1/N) = 1/N - 1/N^2$$

Complete distribution of Z_1

Theorem

For regular RC4, the probability distribution of Z_1 is as follows,

$$\Pr(Z_1 = v) = Q_v + \sum_{X \in \mathcal{L}_v} \sum_{Y \in \mathcal{T}_{v,X}} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X+Y] = v),$$

$$\textit{with} \quad Q_v = \begin{cases} \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = 0), & \textit{if } v = 0; \\ \Pr(S_0[1] = 0 \ \wedge \ S_0[0] = 1), & \textit{if } v = 1; \\ \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = v) & \\ \qquad + \Pr(S_0[1] = v \ \wedge \ S_0[v] = 0) & \\ \qquad + \Pr(S_0[1] = 1 - v \ \wedge \ S_0[1-v] = v), & \textit{otherwise.} \end{cases}$$

where $v \in \{0, \ldots, N-1\}$, $\mathcal{L}_v = \{0, 1, \ldots, N-1\} \setminus \{1, v\}$,
$\mathcal{T}_{v,X} = \{0, 1, \ldots, N-1\} \setminus \{0, X, 1-X, v\}$.

Complete distribution of Z_1

Idea for the proof.

One may write

$$\begin{aligned} Z_1 &= S_1[S_1[i_1] + S_1[j_1]] = S_1[S_0[j_1] + S_0[i_1]] \\ &= S_1[S_0[S_0[1]] + S_0[1]] = S_1[Y + X], \text{ where } X = S_0[1], Y = S_0[X] \end{aligned}$$

Complete distribution of Z_1

Idea for the proof.

One may write

$$\begin{aligned} Z_1 &= S_1[S_1[i_1] + S_1[j_1]] = S_1[S_0[j_1] + S_0[i_1]] \\ &= S_1[S_0[S_0[1]] + S_0[1]] = S_1[Y + X], \text{ where } X = S_0[1], Y = S_0[X] \end{aligned}$$

and thus compute

$$\Pr(Z_1 = v) = \sum_{X=0}^{N-1} \sum_{Y=0}^{N-1} \Pr(S_0[1] = X \wedge S_0[X] = Y \wedge S_1[X+Y] = v).$$

Complete distribution of Z_1

Idea for the proof.

$$\Pr(Z_1 = v) = \sum_{X=0}^{N-1} \sum_{Y=0}^{N-1} \Pr(S_0[1] = X \wedge S_0[X] = Y \wedge S_1[X+Y] = v).$$

We have a known distribution for $S_0[u] = v$ (Mantin, 2001).
Thus the goal is to reduce the term $S_1[X + Y]$ to the state S_0.

Complete distribution of Z_1

Idea for the proof.

$$\Pr(Z_1 = v) = \sum_{X=0}^{N-1} \sum_{Y=0}^{N-1} \Pr(S_0[1] = X \wedge S_0[X] = Y \wedge S_1[X+Y] = v).$$

We have a known distribution for $S_0[u] = v$ (Mantin, 2001).
Thus the goal is to reduce the term $S_1[X + Y]$ to the state S_0.

Note that

- S_1 is different from S_0 in at most two places, $i_1 = 1$ and $j_1 = X$.
- Special cases for $X + Y = 1$ and $X + Y = X$ must be considered.

Complete distribution of Z_1

Idea for the proof.

Special cases depending on X, Y

- $X + Y = 1$ if and only if $Y = 1 - X$, which implies

$$Z_1 = S_1[1] = S_1[i_1] = S_0[j_1] = S_0[X] = Y = 1 - X$$

- $X + Y = X$ if and only if $Y = 0$, which implies

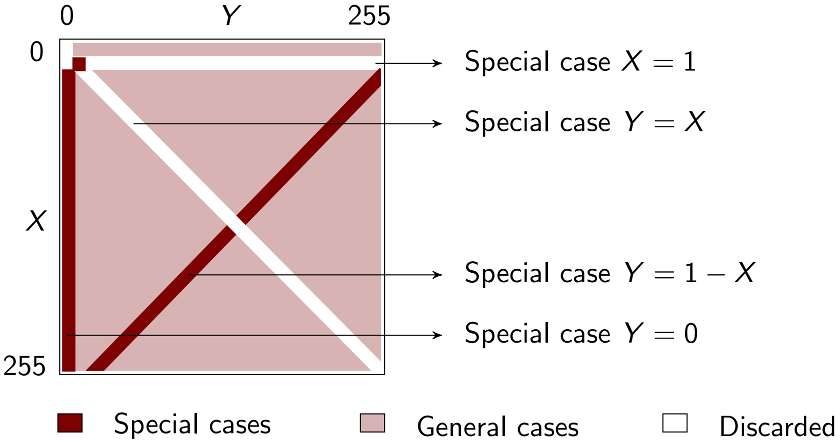
$$Z_1 = S_1[X] = S_1[j_1] = S_0[i_1] = S_0[1] = X$$

- $X = 1$ if and only if $Y = X$, which implies

$$Z_1 = S_1[X + Y] = S_0[X + Y] = S_0[1 + 1] = S_0[2]$$

Complete distribution of Z_1

Idea for the proof.



Complete distribution of Z_1

Idea for the proof.

$$\begin{aligned}
\Pr(Z_1 = v) &= \sum_{X=0}^{N-1} \Pr(S_0[1] = X \ \wedge \ S_0[X] = 1 - X \ \wedge \ 1 - X = v) \\
&+ \sum_{X=0}^{N-1} \Pr(S_0[1] = X \ \wedge \ S_0[X] = 0 \ \wedge \ X = v) \\
&+ \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = v) \\
&+ \sum_{X \neq 1} \sum_{Y \neq 0, X, 1-X} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X+Y] = v).
\end{aligned}$$

Complete distribution of Z_1

Idea for the proof.

$$
\begin{aligned}
\Pr(Z_1 = v) = & \sum_{X=0}^{N-1} \Pr(S_0[1] = X \ \wedge \ S_0[X] = 1 - X \ \wedge \ 1 - X = v) \\
& + \sum_{X=0}^{N-1} \Pr(S_0[1] = X \ \wedge \ S_0[X] = 0 \ \wedge \ X = v) \\
& + \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = v) \\
& + \sum_{X \neq 1} \sum_{Y \neq 0, X, 1-X} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X+Y] = v).
\end{aligned}
$$

The first summation term reduces to a single point $(X = 1 - v, Y = v)$, as we fix $1 - X = v$ and $Y = 1 - X$.

Complete distribution of Z_1

Idea for the proof.

$$
\begin{aligned}
\Pr(Z_1 = v) = {} & \Pr(S_0[1] = 1 - v \ \wedge \ S_0[1-v] = v) \\
& + \sum_{X=0}^{N-1} \Pr(S_0[1] = X \ \wedge \ S_0[X] = 0 \ \wedge \ X = v) \\
& + \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = v) \\
& + \sum_{X \neq 1} \sum_{Y \neq 0, X, 1-X} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X+Y] = v).
\end{aligned}
$$

The second summation, similarly, reduces to point $(X = v, Y = 0)$.

Complete distribution of Z_1

Idea for the proof.

$$\begin{aligned}
\Pr(Z_1 = v) = {} & \Pr(S_0[1] = 1 - v \ \wedge\ S_0[1 - v] = v) \\
& + \Pr(S_0[1] = v \ \wedge\ S_0[v] = 0) \\
& + \Pr(S_0[1] = 1 \ \wedge\ S_0[2] = v) \\
& + \sum_{X \neq 1} \sum_{Y \neq 0, X, 1-X} \Pr(S_0[1] = X \ \wedge\ S_0[X] = Y \ \wedge\ S_0[X + Y] = v).
\end{aligned}$$

Finally, we get two impossible conditions on the double summation: $(X = v, Y \neq 0) \Rightarrow (Z_1 \neq v)$ and $(X \neq 1 - v, Y = v) \Rightarrow (Z_1 \neq v)$.

Complete distribution of Z_1

Idea for the proof.

$$\begin{aligned}\Pr(Z_1 = v) = &\Pr(S_0[1] = 1 - v \ \wedge \ S_0[1-v] = v) \\ &+ \Pr(S_0[1] = v \ \wedge \ S_0[v] = 0) \\ &+ \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = v) \\ &+ \sum_{X \neq 1, v} \ \sum_{Y \neq 0, X, 1-X, v} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X+Y] = v).\end{aligned}$$

Complete distribution of Z_1

Idea for the proof.

$$\begin{aligned}\Pr(Z_1 = v) = {} & \Pr(S_0[1] = 1 - v \ \wedge \ S_0[1 - v] = v) \\ & + \Pr(S_0[1] = v \ \wedge \ S_0[v] = 0) \\ & + \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = v) \\ & + \sum_{X \neq 1, v} \ \sum_{Y \neq 0, X, 1-X, v} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X + Y] = v).\end{aligned}$$

- $v = 0$ reduces the first three terms to $\Pr(S_0[1] = 1 \ \wedge \ S_0[2] = 0)$.

Complete distribution of Z_1

Idea for the proof.

$$\begin{aligned}
\Pr(Z_1 = v) = {} & \Pr(S_0[1] = 1 - v \ \wedge \ S_0[1 - v] = v) \\
& + \Pr(S_0[1] = v \ \wedge \ S_0[v] = 0) \\
& + \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = v) \\
& + \sum_{X \neq 1, v} \ \sum_{Y \neq 0, X, 1-X, v} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X + Y] = v).
\end{aligned}$$

- $v = 0$ reduces the first three terms to $\Pr(S_0[1] = 1 \ \wedge \ S_0[2] = 0)$.
- $v = 1$ reduces the first three terms to $\Pr(S_0[1] = 0 \ \wedge \ S_0[0] = 1)$.

Complete distribution of Z_1

Idea for the proof.

$$\begin{aligned}
\Pr(Z_1 = v) = {} & \Pr(S_0[1] = 1 - v \ \wedge \ S_0[1 - v] = v) \\
& + \Pr(S_0[1] = v \ \wedge \ S_0[v] = 0) \\
& + \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = v) \\
& + \sum_{X \neq 1, v} \ \sum_{Y \neq 0, X, 1-X, v} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X + Y] = v).
\end{aligned}$$

- $v = 0$ reduces the first three terms to $\Pr(S_0[1] = 1 \ \wedge \ S_0[2] = 0)$.
- $v = 1$ reduces the first three terms to $\Pr(S_0[1] = 0 \ \wedge \ S_0[0] = 1)$.
- $v \neq 0, 1$ keeps all the first three terms intact.

Complete distribution of Z_1

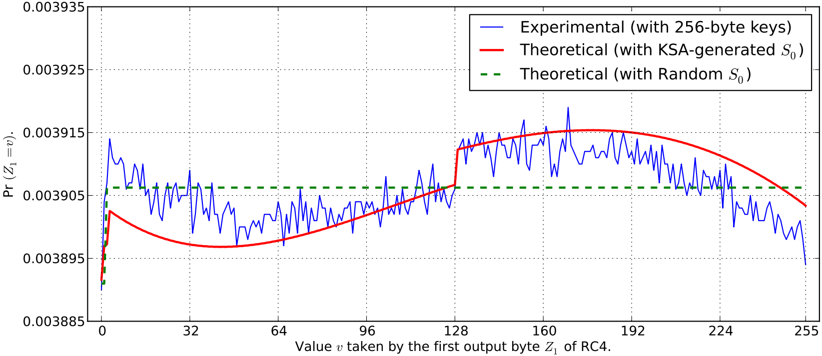
Hence the final expression

$$\Pr(Z_1 = v) = Q_v + \sum_{X \in \mathcal{L}_v} \sum_{Y \in \mathcal{T}_{v,X}} \Pr(S_0[1] = X \ \wedge \ S_0[X] = Y \ \wedge \ S_0[X+Y] = v),$$

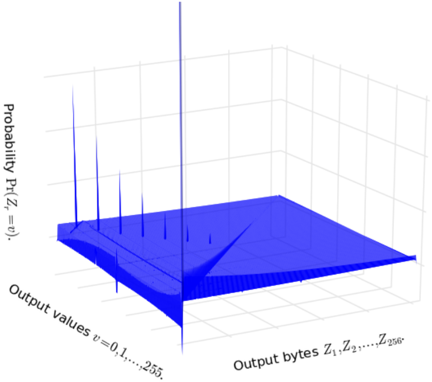
$$\text{with} \quad Q_v = \begin{cases} \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = 0), & \text{if } v = 0; \\ \Pr(S_0[1] = 0 \ \wedge \ S_0[0] = 1), & \text{if } v = 1; \\ \Pr(S_0[1] = 1 \ \wedge \ S_0[2] = v) & \\ \quad + \Pr(S_0[1] = v \ \wedge \ S_0[v] = 0) & \\ \quad + \Pr(S_0[1] = 1 - v \ \wedge \ S_0[1-v] = v), & \text{otherwise.} \end{cases}$$

where $v \in \{0, \ldots, N-1\}$, $\mathcal{L}_v = \{0, 1, \ldots, N-1\} \setminus \{1, v\}$,
$\mathcal{T}_{v,X} = \{0, 1, \ldots, N-1\} \setminus \{0, X, 1-X, v\}$.

Complete distribution of Z_1

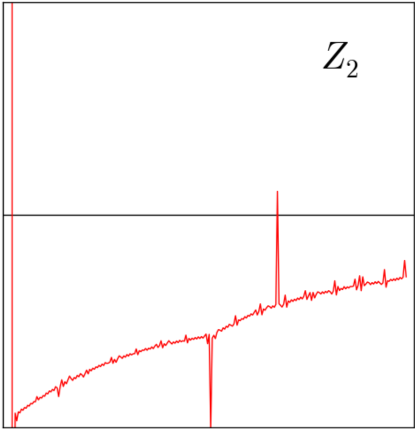


Observed by Mironov in 2002. Proved by SMPS in 2013.



Other initial bytes of RC4

Non-randomness in Z_2



$$\Pr(Z_2 = v)$$
$$v = 0, 1, \ldots, 255$$

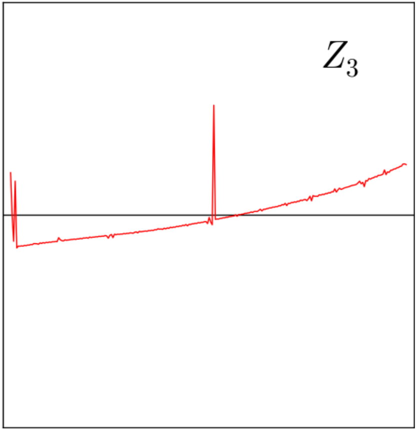
Major biases

$$\Pr(Z_2 = 0) \approx \frac{2}{N}$$

$$\Pr(Z_2 = 129) \approx \frac{1}{N} - \frac{2}{N^2}$$

$$\Pr(Z_2 = 172) \approx \frac{1}{N} + \frac{0.2}{N^2}$$

Non-randomness in Z_3



$$\Pr(Z_3 = v)$$
$$v = 0, 1, \ldots, 255$$

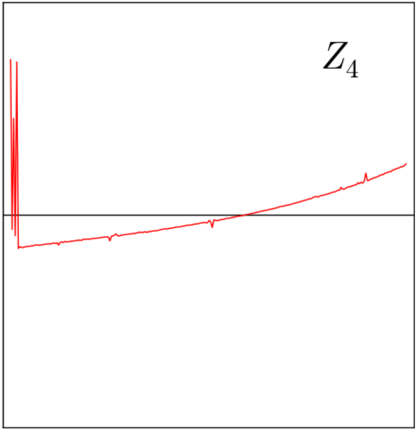
Major biases

$$\Pr(Z_3 = 0) \approx \frac{1}{N} + \frac{0.3}{N^2}$$

$$\Pr(Z_3 = 3) \approx \frac{1}{N} + \frac{0.3}{N^2}$$

$$\Pr(Z_3 = 131) \approx \frac{1}{N} + \frac{2}{N^2}$$

Non-randomness in Z_4



$$\Pr(Z_4 = v)$$
$$v = 0, 1, \ldots, 255$$

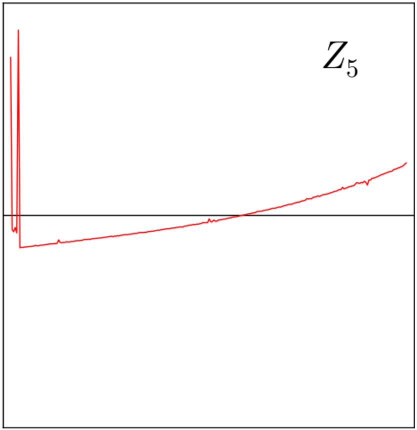
Major biases

$$\Pr(Z_4 = 0) \approx \frac{1}{N} + \frac{1}{N^2}$$

$$\Pr(Z_4 = 4) \approx \frac{1}{N} + \frac{1}{N^2}$$

$$\Pr(Z_4 = 2) \approx \frac{1}{N} + \frac{0.8}{N^2}$$

Non-randomness in Z_5



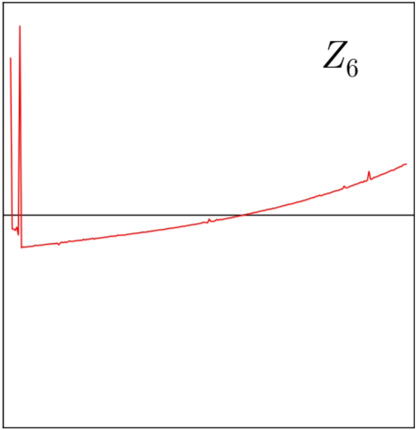
$$\Pr(Z_5 = v)$$
$$v = 0, 1, \ldots, 255$$

Major biases

$$\Pr(Z_5 = 0) \approx \frac{1}{N} + \frac{1}{N^2}$$

$$\Pr(Z_5 = 5) \approx \frac{1}{N} + \frac{1}{N^2}$$

Non-randomness in Z_6



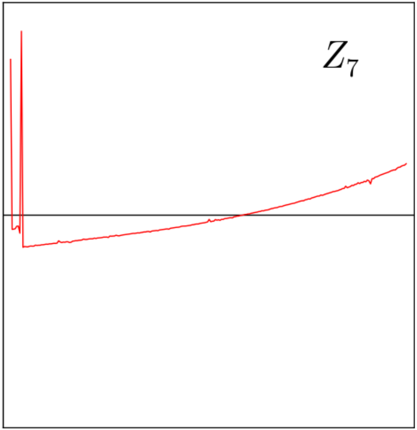
$$\Pr(Z_6 = v)$$
$$v = 0, 1, \ldots, 255$$

Major biases

$$\Pr(Z_6 = 0) \approx \frac{1}{N} + \frac{1}{N^2}$$

$$\Pr(Z_6 = 6) \approx \frac{1}{N} + \frac{1}{N^2}$$

Non-randomness in Z_7



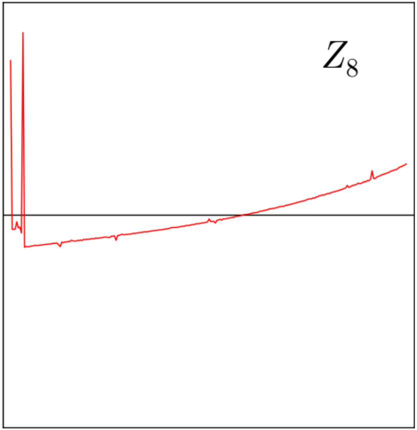
$$\Pr(Z_7 = v)$$
$$v = 0, 1, \ldots, 255$$

Major biases

$$\Pr(Z_7 = 0) \approx \frac{1}{N} + \frac{1}{N^2}$$

$$\Pr(Z_7 = 7) \approx \frac{1}{N} + \frac{1}{N^2}$$

Non-randomness in Z_8



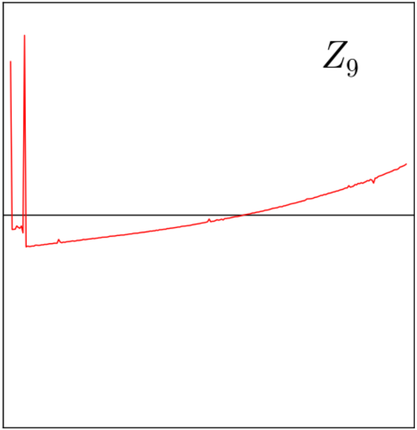
$$\Pr(Z_8 = v)$$
$$v = 0, 1, \ldots, 255$$

Major biases

$$\Pr(Z_8 = 0) \approx \frac{1}{N} + \frac{1}{N^2}$$

$$\Pr(Z_8 = 8) \approx \frac{1}{N} + \frac{1}{N^2}$$

Non-randomness in Z_9



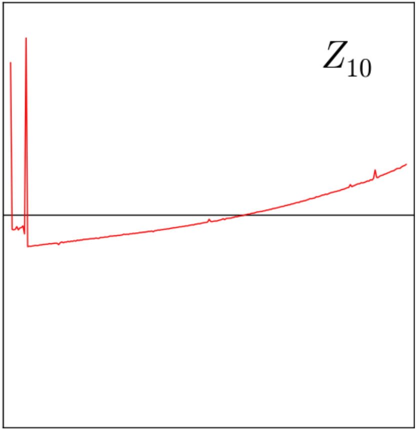
$$\Pr(Z_9 = v)$$
$$v = 0, 1, \ldots, 255$$

Major biases

$$\Pr(Z_9 = 0) \approx \frac{1}{N} + \frac{1}{N^2}$$

$$\Pr(Z_9 = 9) \approx \frac{1}{N} + \frac{1}{N^2}$$

Non-randomness in Z_{10}



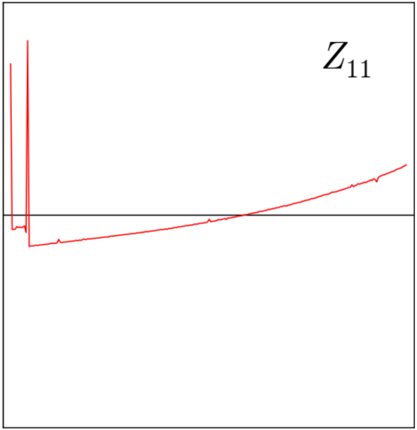
$$\Pr(Z_{10} = v)$$
$$v = 0, 1, \ldots, 255$$

Major biases

$$\Pr(Z_{10} = 0) \approx \frac{1}{N} + \frac{1}{N^2}$$

$$\Pr(Z_{10} = 10) \approx \frac{1}{N} + \frac{1}{N^2}$$

Non-randomness in Z_{11}



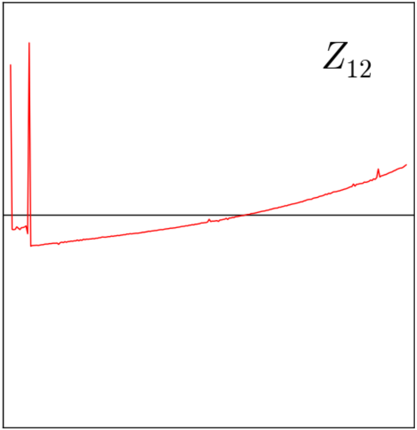
$$\Pr(Z_{11} = v)$$
$$v = 0, 1, \ldots, 255$$

Major biases

$$\Pr(Z_{11} = 0) \approx \frac{1}{N} + \frac{1}{N^2}$$

$$\Pr(Z_{11} = 11) \approx \frac{1}{N} + \frac{1}{N^2}$$

Non-randomness in Z_{12}



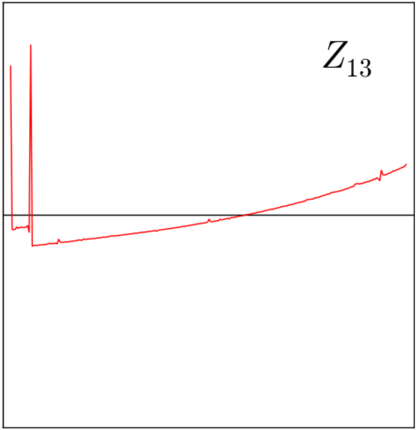
$\Pr(Z_{12} = v)$
$v = 0, 1, \ldots, 255$

Major biases

$\Pr(Z_{12} = 0) \approx \frac{1}{N} + \frac{1}{N^2}$

$\Pr(Z_{12} = 12) \approx \frac{1}{N} + \frac{1}{N^2}$

Non-randomness in Z_{13}



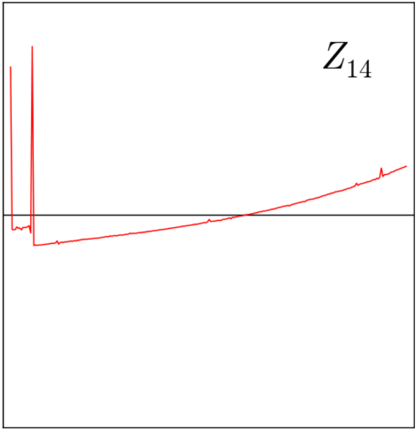
$$\Pr(Z_{13} = v)$$
$$v = 0, 1, \ldots, 255$$

Major biases

$$\Pr(Z_{13} = 0) \approx \frac{1}{N} + \frac{1}{N^2}$$

$$\Pr(Z_{13} = 13) \approx \frac{1}{N} + \frac{1}{N^2}$$

Non-randomness in Z_{14}



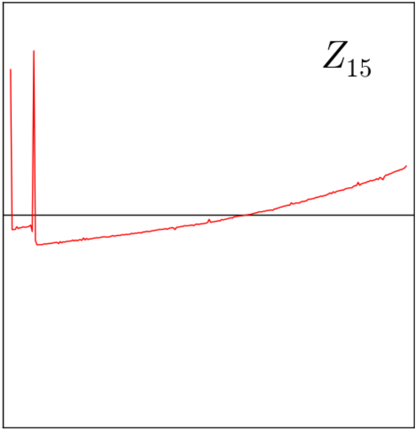
$$\Pr(Z_{14} = v)$$
$$v = 0, 1, \ldots, 255$$

Major biases

$$\Pr(Z_{14} = 0) \approx \frac{1}{N} + \frac{1}{N^2}$$

$$\Pr(Z_{14} = 14) \approx \frac{1}{N} + \frac{1}{N^2}$$

Non-randomness in Z_{15}



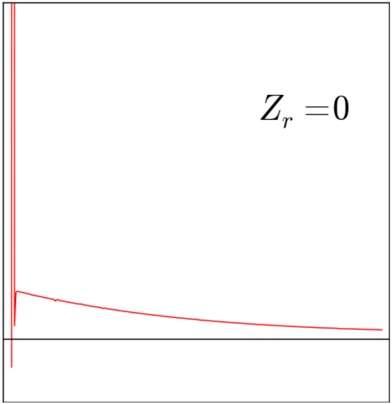
$\Pr(Z_{15} = v)$
$v = 0, 1, \ldots, 255$

Major biases

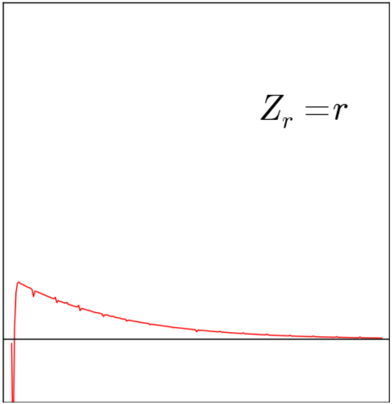
$\Pr(Z_{15} = 0) \approx \frac{1}{N} + \frac{1}{N^2}$

$\Pr(Z_{15} = 15) \approx \frac{1}{N} + \frac{1}{N^2}$

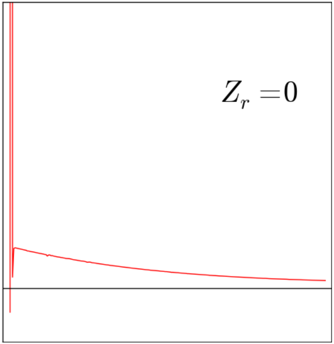
Non-randomness in initial bytes



$\Pr(Z_r = 0)$
$r = 1, 2, \ldots, 255$



$\Pr(Z_r = r)$
$r = 1, 2, \ldots, 255$



Zero-bias of initial bytes

Zero-bias of initial bytes

- Mantin-Shamir discovered and proved the $(Z_2 = 0)$ bias in 2001.
- They claimed there are no biases towards zero for bytes 3 to 255.
- We revisit their work and contradict this claim in 2011.

Zero-bias of initial bytes

- Mantin-Shamir discovered and proved the $(Z_2 = 0)$ bias in 2001.
- They claimed there are no biases towards zero for bytes 3 to 255.
- We revisit their work and contradict this claim in 2011.

Theorem

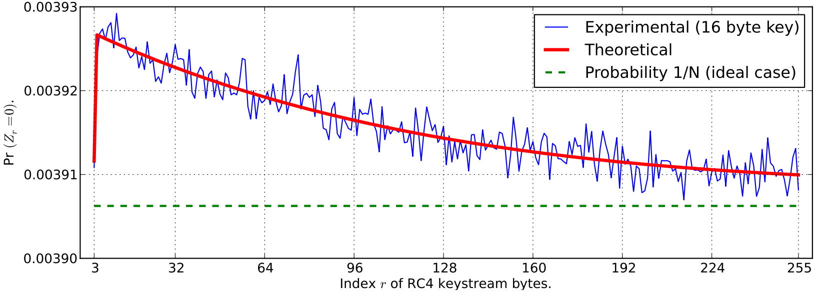
In PRGA rounds $3 \leq r \leq N-1$, probability $\Pr(Z_r = 0)$ is:

$$\Pr(Z_r = 0) \approx \frac{1}{N} + \frac{c_r}{N^2},$$

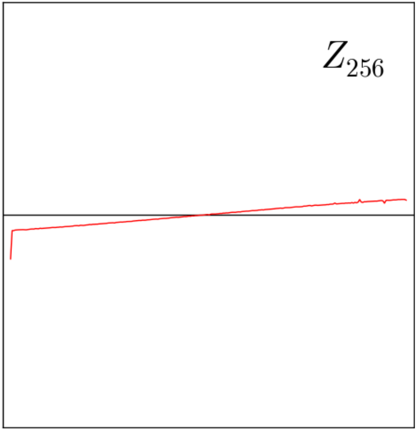
$$\textit{where} \quad c_r = \begin{cases} \frac{N}{N-1}\left(N \cdot \Pr(S_{r-1}[r] = r) - 1\right) - \frac{N-2}{N-1}, & \textit{for } r = 3; \\ \frac{N}{N-1}\left(N \cdot \Pr(S_{r-1}[r] = r) - 1\right), & \textit{otherwise.} \end{cases}$$

Zero-bias of initial bytes

- Mantin-Shamir discovered and proved the $(Z_2 = 0)$ bias in 2001.
- They claimed there are no biases towards zero for bytes 3 to 255.
- We revisit their work and contradict this claim in 2011.



Zero-bias after byte 255

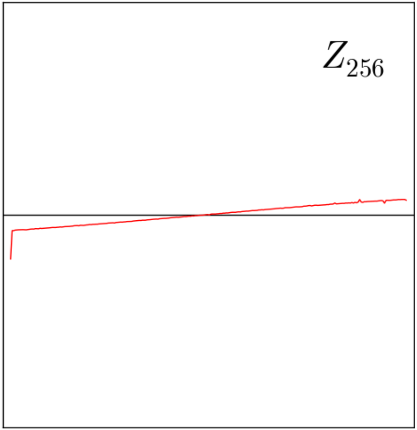


$\Pr(Z_{256} = v)$
$v = 0, 1, \ldots, 255$

We proved

$$\Pr(Z_{256} = 0) \approx \frac{1}{N} - \frac{0.4}{N^2}$$

Zero-bias after byte 255



$\Pr(Z_{256} = v)$
$v = 0, 1, \ldots, 255$

We proved

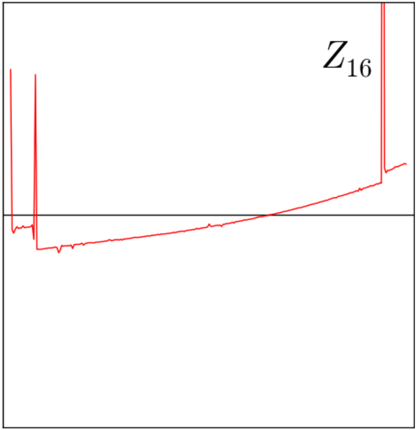
$$\Pr(Z_{256} = 0) \approx \frac{1}{N} - \frac{0.4}{N^2}$$

We also proved

$$\Pr(Z_{257} = 0) \approx \frac{1}{N} + \frac{0.35}{N^2}$$

Something weird happens at the 16-th byte

Strange bias in $(Z_{16} = 240)$



$\Pr(Z_{16} = v)$
$v = 0, 1, \ldots, 255$

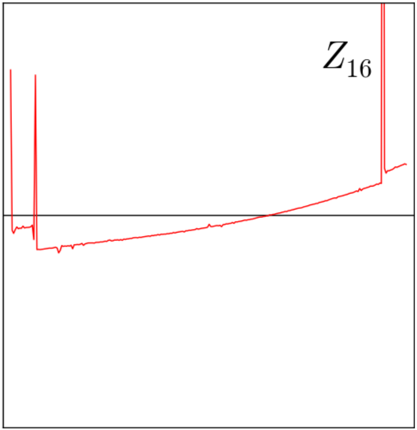
Major biases

$\Pr(Z_{16} = 0) \approx \frac{1}{N} + \frac{1}{N^2}$

$\Pr(Z_{16} = 16) \approx \frac{1}{N} + \frac{1}{N^2}$

$\Pr(Z_{16} = 240) \approx \frac{1}{N} + \frac{9}{N^2}$

Strange bias in $(Z_{16} = 240)$



$\Pr(Z_{16} = v)$
$v = 0, 1, \ldots, 255$

Major biases

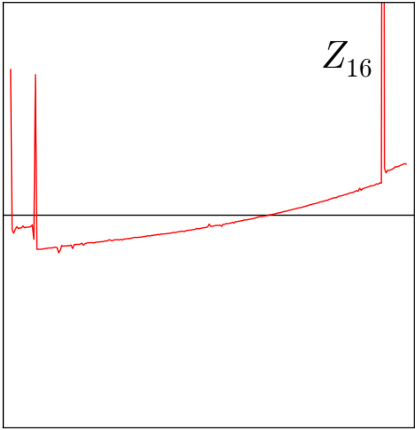
$\Pr(Z_{16} = 0) \approx \frac{1}{N} + \frac{1}{N^2}$

$\Pr(Z_{16} = 16) \approx \frac{1}{N} + \frac{1}{N^2}$

$\Pr(Z_{16} = 240) \approx \frac{1}{N} + \frac{9}{N^2}$

Why 16?

Strange bias in $(Z_{16} = 240)$



$\Pr(Z_{16} = v)$
$v = 0, 1, \ldots, 255$

Major biases

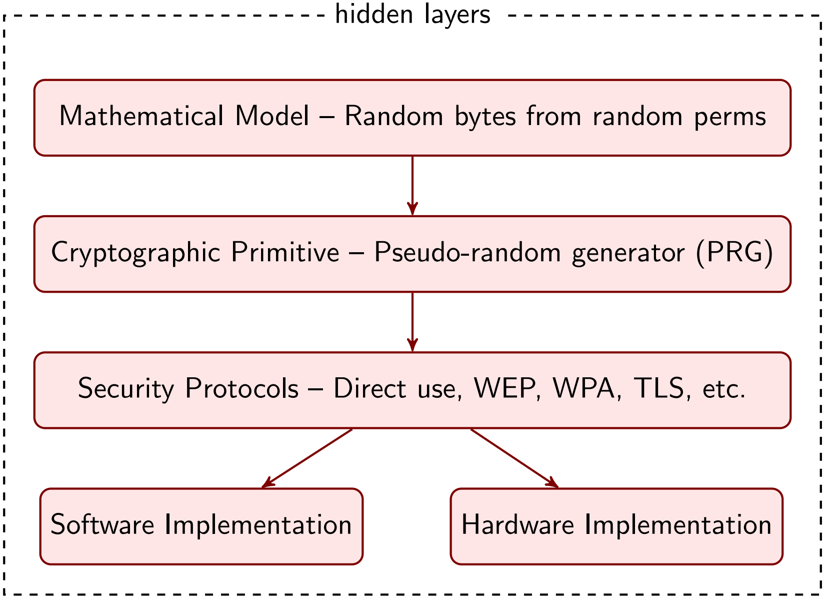
$\Pr(Z_{16} = 0) \approx \frac{1}{N} + \frac{1}{N^2}$

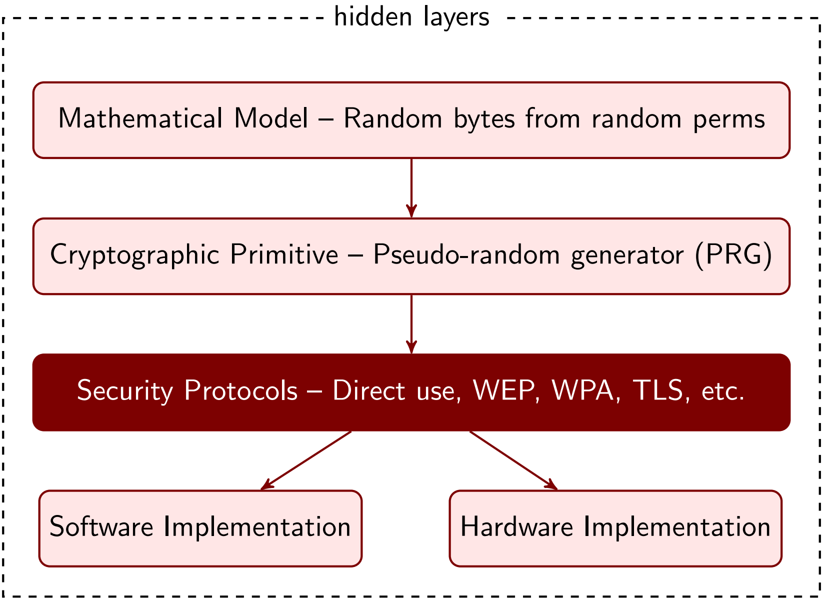
$\Pr(Z_{16} = 16) \approx \frac{1}{N} + \frac{1}{N^2}$

$\Pr(Z_{16} = 240) \approx \frac{1}{N} + \frac{9}{N^2}$

Why 16?

$240 \equiv -16$


hidden layers
Mathematical Model – Random bytes from random perms
Cryptographic Primitive – Pseudo-random generator (PRG)
Security Protocols – Direct use, WEP, WPA, TLS, etc.
Software Implementation
Hardware Implementation


hidden layers
Mathematical Model – Random bytes from random perms
Cryptographic Primitive – Pseudo-random generator (PRG)
Security Protocols – Direct use, WEP, WPA, TLS, etc.
Software Implementation
Hardware Implementation

RC4 in Practice

RC4 in practice

For the KSA step $j = j + S[i] + K[i]$, we require 256-byte K array. However in practice, the most typical key-size for RC4 is 128 bits.

RC4 in practice

For the KSA step $j = j + S[i] + K[i]$, we require 256-byte K array. However in practice, the most typical key-size for RC4 is 128 bits.

KEY EXPANSION: $K[i] = \text{RC4KEY}[i \bmod l]$ for $i = 0, 1, 2, \ldots, 255$, where l is the length (in bytes) of the secret key

RC4 in practice

For the KSA step $j = j + S[i] + K[i]$, we require 256-byte K array. However in practice, the most typical key-size for RC4 is 128 bits.

KEY EXPANSION: $K[i] = \text{RC4KEY}[i \bmod l]$ for $i = 0, 1, 2, \ldots, 255$, where l is the length (in bytes) of the secret key

Typical length of the secret key: $l = 128$ bits $= 16$ bytes

Intuition: This keylength of $l = 16$ may have reflected in the Z_{16} bias.

Discovery and proof of keylength-dependent biases

Keylength-dependent distinguisher of RC4

$\Pr(Z_l = -l) > \frac{1}{N} + \frac{1}{N^2}$ for all practical keylengths $l = 5, 6, \ldots, 30$.

Keylength-dependent distinguisher of RC4

$\Pr(Z_l = -l) > \frac{1}{N} + \frac{1}{N^2}$ for all practical keylengths $l = 5, 6, \ldots, 30$.

Theorem

Suppose that l is the length of the secret key of RC4. Then

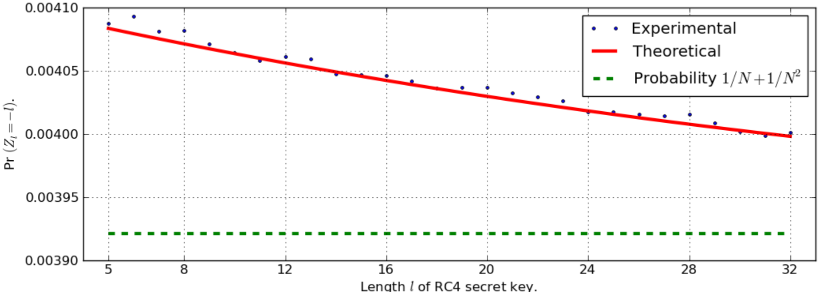
$$\Pr(Z_l = -l) \approx \frac{1}{N^2} + \left(1 - \frac{1}{N^2}\right)\gamma_l + (1 - \delta_l)\frac{1}{N},$$

where $\gamma_l = \frac{1}{N^2}\left(1 - \frac{l+1}{N}\right)\sum_{x=l+1}^{N-1}\left(1 - \frac{1}{N}\right)^x \left(1 - \frac{2}{N}\right)^{x-l}\left(1 - \frac{3}{N}\right)^{N-x+2l-4}$ *and*

$\delta_l = \Pr(S_1[l] = 0)\left(1 - \frac{1}{N}\right)^{l-2} + \sum_{t=2}^{l-1}\sum_{w=0}^{l-t}\frac{\Pr(S_1[t]=0)}{w! \cdot N}\left(\frac{l-t-1}{N}\right)^w \left(1 - \frac{1}{N}\right)^{l-3-w}$.

Keylength-dependent distinguisher of RC4

$\Pr(Z_l = -l) > \frac{1}{N} + \frac{1}{N^2}$ for all practical keylengths $l = 5, 6, \ldots, 32$.



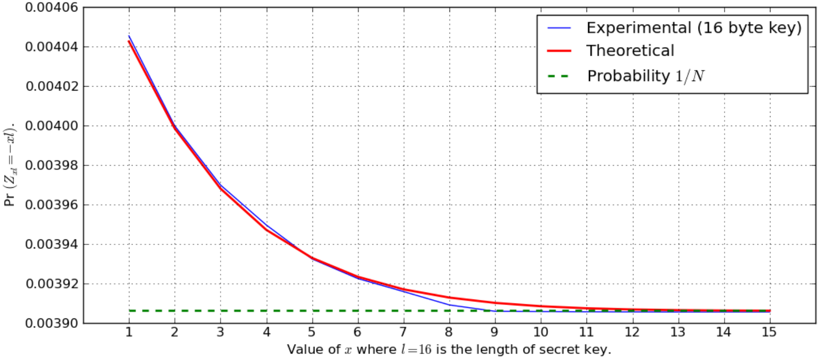
Extended keylength-dependent biases

$\Pr(Z_{xl} = -xl) > \frac{1}{N}$ for $l = 5, 6, \ldots, 32$ and $x = 1, 2, \ldots, \lfloor \frac{N}{l} \rfloor$

Extended keylength-dependent biases

$\Pr(Z_{xl} = -xl) > \frac{1}{N}$ for $l = 5, 6, \ldots, 32$ and $x = 1, 2, \ldots, \lfloor \frac{N}{l} \rfloor$

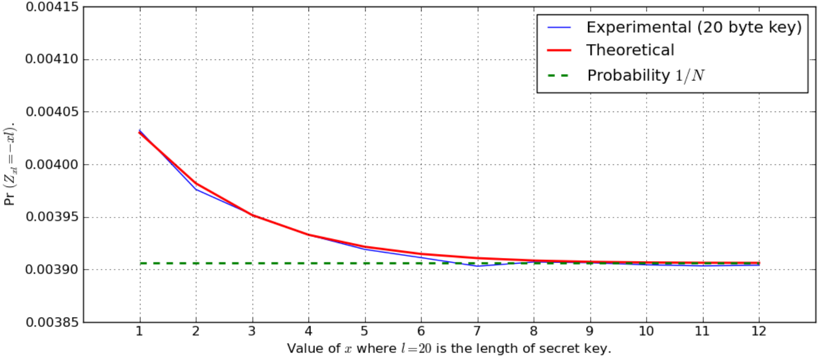
Example for $l = 16$



Extended keylength-dependent biases

$\Pr(Z_{xl} = -xl) > \frac{1}{N}$ for $l = 5, 6, \ldots, 32$ and $x = 1, 2, \ldots, \lfloor \frac{N}{l} \rfloor$

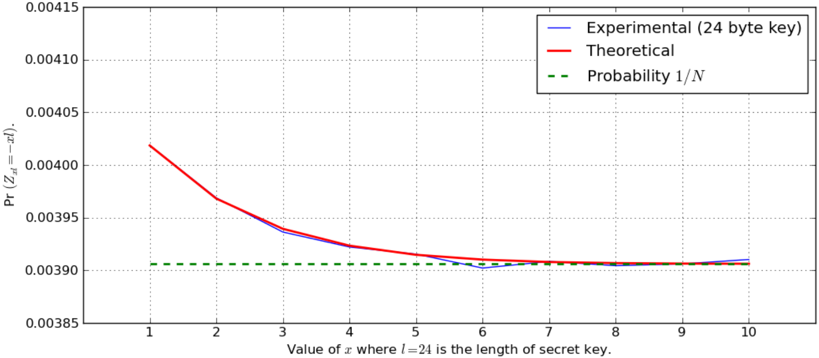
Example for $l = 20$



Extended keylength-dependent biases

$\Pr(Z_{xl} = -xl) > \frac{1}{N}$ for $l = 5, 6, \ldots, 32$ and $x = 1, 2, \ldots, \lfloor \frac{N}{l} \rfloor$

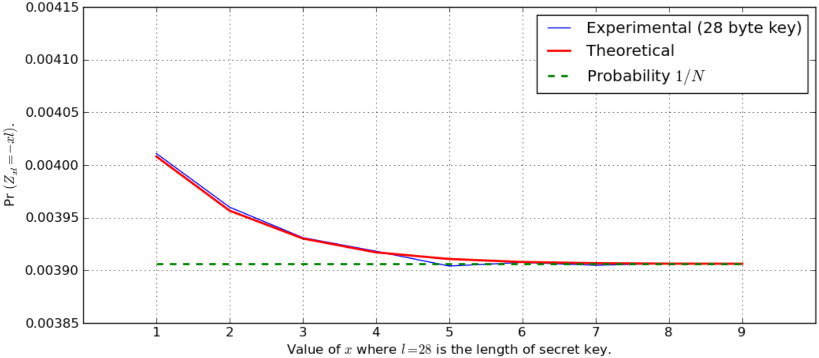
Example for $l = 24$



Extended keylength-dependent biases

$\Pr(Z_{xl} = -xl) > \frac{1}{N}$ for $l = 5, 6, \ldots, 32$ and $x = 1, 2, \ldots, \lfloor \frac{N}{l} \rfloor$

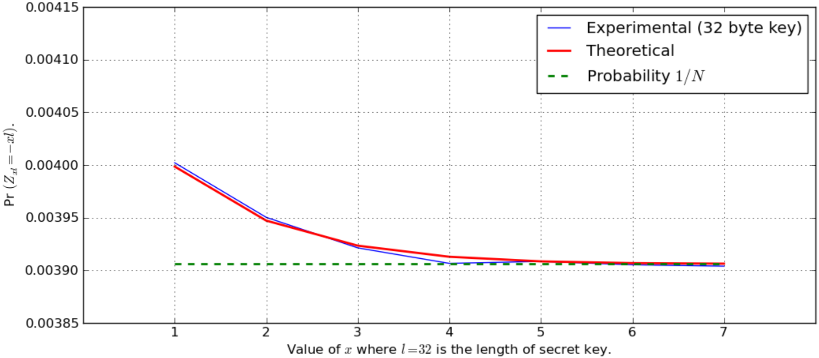
Example for $l = 28$



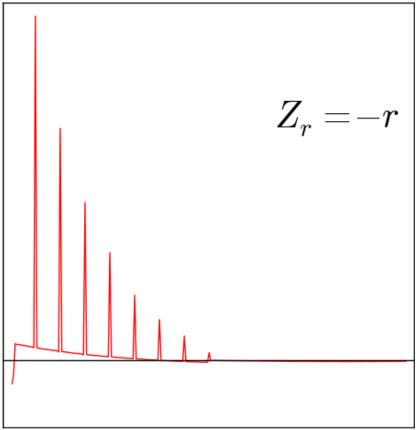
Extended keylength-dependent biases

$\Pr(Z_{xl} = -xl) > \frac{1}{N}$ for $l = 5, 6, \ldots, 32$ and $x = 1, 2, \ldots, \lfloor \frac{N}{l} \rfloor$

Example for $l = 32$



Keylength-dependent biases for $l = 16$



$$\Pr(Z_r = -r)$$
$$r = 1, \ldots, 255$$

Major biases

$$\Pr(Z_{16} = 240) \approx \frac{1}{N} + \frac{9}{N^2}$$

$$\Pr(Z_{32} = 224) \approx \frac{1}{N} + \frac{6}{N^2}$$

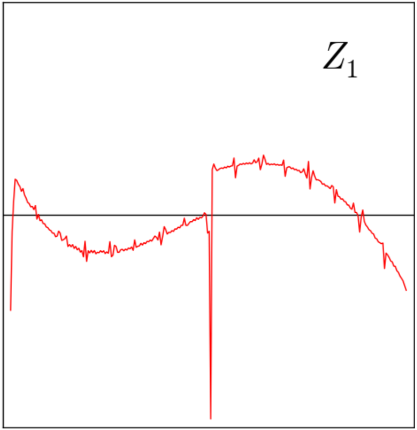
$$\Pr(Z_{48} = 208) \approx \frac{1}{N} + \frac{4}{N^2}$$

$$\Pr(Z_{64} = 192) \approx \frac{1}{N} + \frac{3}{N^2}$$

$$\Pr(Z_{80} = 176) \approx \frac{1}{N} + \frac{2}{N^2}$$

Keylength affects Z_1 too

Keylength-dependence in Z_1



$\Pr(Z_1 = v)$
$v = 0, 1, \ldots, 255$

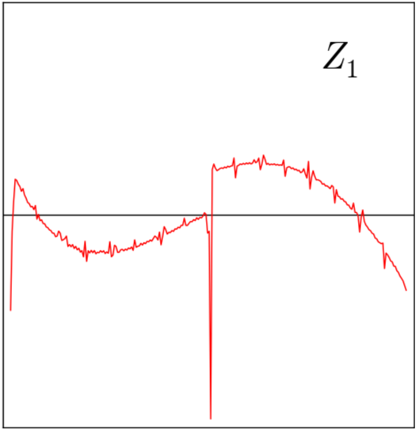
Major biases

Sinusoidal distribution

$\Pr(Z_1 = 0) \approx \frac{1}{N} - \frac{1}{N^2}$

$\Pr(Z_1 = 129) \approx \frac{1}{N} - \frac{2}{N^2}$

Keylength-dependence in Z_1



$\Pr(Z_1 = v)$
$v = 0, 1, \ldots, 255$

Major biases

Sinusoidal distribution

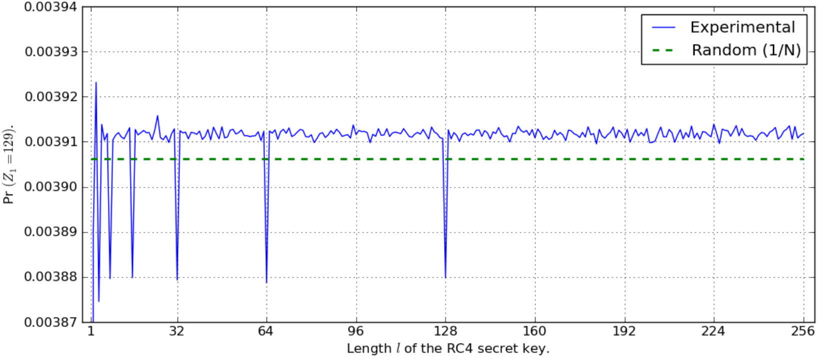
$\Pr(Z_1 = 0) \approx \frac{1}{N} - \frac{1}{N^2}$

$\boxed{\Pr(Z_1 = 129) \approx \frac{1}{N} - \frac{2}{N^2}}$

For $l = 16$, not for $l = 256$

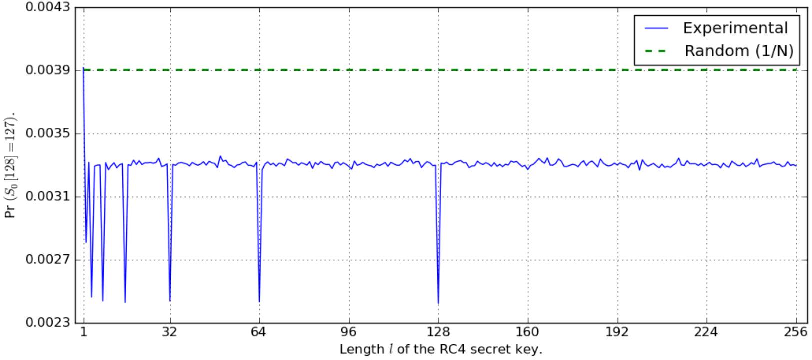
Keylength-dependence in Z_1

Bias at $(Z_1 = 129)$ is present only for $l = 2, 4, 8, 16, 32, 64, 128$



Keylength-dependence in S_0

Bias at $(S_0[128] = 127)$ is present only for $l = 2, 4, 8, 16, 32, 64, 128$



Keylength-dependence in S_0

$(S_0[128] = 127)$ bias for $l = 16$ was known as an *anomaly* since 2001. We prove the general result in this direction in 2013.

Keylength-dependence in S_0

$(S_0[128] = 127)$ bias for $l = 16$ was known as an *anomaly* since 2001. We prove the general result in this direction in 2013.

Theorem

In practical RC4 with $N = 256$,

$$\Pr(S_0[128] = 127) \approx 0.63/N,$$

if and only if l is a non-trivial factor of $N = 256$.

Keylength-dependence in S_0

$(S_0[128] = 127)$ bias for $l = 16$ was known as an *anomaly* since 2001. We prove the general result in this direction in 2013.

Theorem
In practical RC4 with $N = 256$*,*

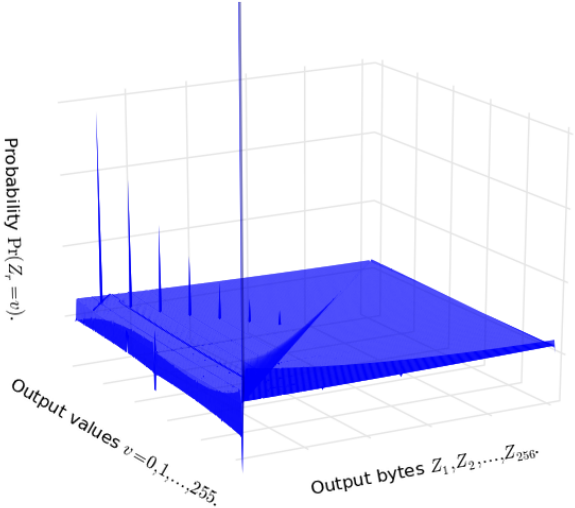
$$\Pr(S_0[128] = 127) \approx 0.63/N,$$

if and only if l is a non-trivial factor of $N = 256$*.*

Intuition for the proof: The calculation for $\Pr(S_0[128] = 127)$ behaves differently if $K[128] = K[0]$ after key expansion; this happens with certainty if and only if $l = 2, 4, 8, 16, 32, 64, 128$.

Practical implication of initial-byte biases

RC4 becomes weak against broadcast attack on initial plaintext bytes!



Recent plaintext-recovery attacks

Our result on biases in $(Z_r = 0)$ first opened the possibility for recovery of plaintext bytes other than the second one.

Recent plaintext-recovery attacks

Our result on biases in $(Z_r = 0)$ first opened the possibility for recovery of plaintext bytes other than the second one.

MPS 2011: Recovery of $P_3, P_4, \ldots, P_{255}$ from $\Omega(N^3)$ ciphertexts.

Recent plaintext-recovery attacks

Our result on biases in $(Z_r = 0)$ first opened the possibility for recovery of plaintext bytes other than the second one.

MPS 2011: Recovery of $P_3, P_4, \ldots, P_{255}$ from $\Omega(N^3)$ ciphertexts.

Isobe et al., 2013

- Recovery of all initial bytes using a chosen set of biases.

Recent plaintext-recovery attacks

Our result on biases in $(Z_r = 0)$ first opened the possibility for recovery of plaintext bytes other than the second one.

MPS 2011: Recovery of $P_3, P_4, \ldots, P_{255}$ from $\Omega(N^3)$ ciphertexts.

Isobe et al., 2013

- Recovery of all initial bytes using a chosen set of biases.

AlFardan et al., 2013

- Recovery of all initial bytes using all initial byte biases.
- Broadcast attack on TLS using the same technique.

Discard all problematic initial output bytes!

Long-term bias in RC4

Golic proved a bitwise correlation between Z_r and Z_{r+2} in 1997.
We prove a new periodic bytewise correlation between Z_r and Z_{r+2}.

Long-term bias in RC4

Golic proved a bitwise correlation between Z_r and Z_{r+2} in 1997.
We prove a new periodic bytewise correlation between Z_r and Z_{r+2}.

Theorem

Suppose that the permutation S_{wN} is truly random, then for $w > 0$,

$$\Pr(Z_{wN+2} = 0 \ \wedge \ Z_{wN} = 0) \approx \frac{1}{N^2} + \frac{1}{N^3}.$$

Long-term bias in RC4

Golic proved a bitwise correlation between Z_r and Z_{r+2} in 1997.
We prove a new periodic bytewise correlation between Z_r and Z_{r+2}.

Theorem

Suppose that the permutation S_{wN} is truly random, then for $w > 0$,

$$\Pr(Z_{wN+2} = 0 \ \wedge \ Z_{wN} = 0) \approx \frac{1}{N^2} + \frac{1}{N^3}.$$

This is the first long-term byte-wise correlation (periodic) to be observed between two non-consecutive bytes.

Biases related to the state-variables

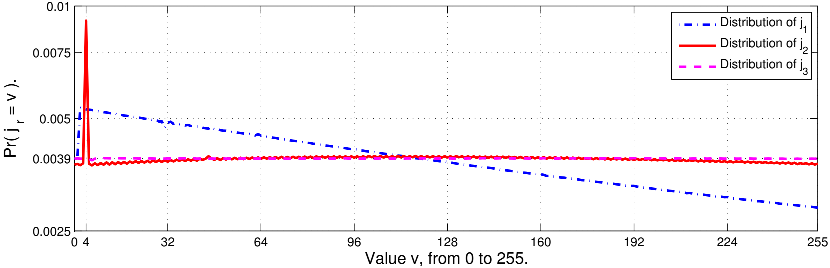
State-dependent biases

Observed by SVV in 2010, proved by SMPS in 2011.

Type of Bias	Label by SVV'10	Biases proved
	"New_004"	$j_2 + S_2[j_2] = S_2[i_2] + Z_2$
Specific	"New_noz_007"	$j_2 + S_2[j_2] = 6$
Initial Rounds	"New_noz_009"	$j_2 + S_2[j_2] = S_2[i_2]$
	"New_noz_014"	$j_1 + S_1[i_1] = 2$
All Rounds	"New_noz_001"	$j_r + S_r[i_r] = i_r + S_r[j_r]$
(r-independent)	"New_noz_002"	$j_r + S_r[j_r] = i_r + S_r[i_r]$
All Initial	"New_000"	$S_r[t_r] = t_r$
Rounds	"New_noz_004"	$S_r[i_r] = j_r$
(r-dependent)	"New_noz_006"	$S_r[j_r] = i_r$

Non-randomness of index j

We characterized the non-randomness in index j
and in the process, discovered a new bias in $(j_2 = 4)$.



Index j behaves random from onwards j_3.

Glimpse in RC4

We exploited the bias in $(j_2 = 4)$ to get a short-term glimpse.

$$\Pr\left(S_2[2] = 4 - Z_2\right) \approx \frac{1}{N} + \frac{4/3}{N^2}.$$

Glimpse in RC4

We exploited the bias in $(j_2 = 4)$ to get a short-term glimpse.

$$\Pr(S_2[2] = 4 - Z_2) \approx \frac{1}{N} + \frac{4/3}{N^2}.$$

The best existing long-term glimpse was by Jenkins in 1996.

$$\Pr(S_r[j_r] = i_r - Z_r) = \Pr(S_r[i_r] = j_r - Z_r) \approx \frac{2}{N}$$

Glimpse in RC4

We exploited the bias in $(j_2 = 4)$ to get a short-term glimpse.

$$\Pr(S_2[2] = 4 - Z_2) \approx \frac{1}{N} + \frac{4/3}{N^2}.$$

The best existing long-term glimpse was by Jenkins in 1996.

$$\Pr(S_r[j_r] = i_r - Z_r) = \Pr(S_r[i_r] = j_r - Z_r) \approx \frac{2}{N}$$

We identified the proved a new long-term glimpse in 2013.

$$\Pr(S_r[r+1] = N - 1 \mid Z_{r+1} = Z_r \ \wedge \ Z_{r+1} = r + 2) \ \approx \ \frac{3}{N}$$

Contributions in RC4 Analysis

Contributions in RC4 Analysis

Settling long-standing open problems	Ref.
1. Keylength dependent anomaly	Mantin, 2001
2. Long-term conditional glimpse	Jenkins, 1996
3. Distribution of Z_1	Mironov, 2002
4. Zero-bias of bytes $Z_3, \ldots, Z_{255}$	MS, 2001
5. Long-term bias in non-consecutive bytes	Golic, 1997

Contributions in RC4 Analysis

Providing theoretical validation of practical attacks	Ref.
1. Proving biases used in WEP and WPA attacks	SVV, 2010
2. Proving biases used in recent TLS attacks	ABPPS, 2013

Contributions in RC4 Analysis

Providing theoretical validation of practical attacks	Ref.
1. Proving biases used in WEP and WPA attacks	SVV, 2010
2. Proving biases used in recent TLS attacks	ABPPS, 2013

Initiating new directions in RC4 analysis	Ref.
1. Keylength-dependent biases in RC4	SMPS, 2013
2. Keylength-dependence in Z_1 bias	SSPM, 2013

Part II

Implementation of RC4

Motivation for this Work

"In how many clocks a byte can be generated in RC4 PRGA?"

Most common approach

- 1 cycle for increment/computation of indices i, j
- 1 cycle for swapping the values $S[i]$ and $S[j]$
- 1 cycle for reading the Z value from S-array

MOTIVATION: Can we get a better throughput?

Design 1 – Loop unrolling

"One Byte per Clock throughput for RC4 PRGA"

$$3 \longrightarrow 1$$

- N bytes of output in $N + 2$ clock cycles
- Completion of RC4 KSA in 257 clock cycles
- Asymptotically *'one byte per clock cycle'*

Design 1 – Loop unrolling

"Combine two rounds of RC4 PRGA"

Steps	First Loop	Second Loop
1	$i_1 = i_0 + 1$	$i_2 = i_1 + 1 = i_0 + 2$
2	$j_1 = j_0 + S_0[i_1]$	$j_2 = j_1 + S_1[i_2] = j_0 + S_0[i_1] + S_1[i_2]$
3	Swap $S_0[i_1] \leftrightarrow S_0[j_1]$	Swap $S_1[i_2] \leftrightarrow S_1[j_2]$
4	$Z_1 = S_1[S_0[i_1] + S_0[j_1]]$	$Z_2 = S_2[S_1[i_2] + S_1[j_2]]$

- What if the indices overlap? (e.g., $j_1 = i_2$)
- What about the ordering of *Swap* and *Output*?

Design 1 – Loop unrolling

	Stage 1	Stage 2	Stage 3
Cycle 1	$i_1 = i_0 + 1;$ $j_1 = j_0 + S_0[i_1];$ $i_2 = i_1 + 1;$ $j_2 = j_1 + S_1[i_2];$		
Cycle 2		Swap $S_0[i_1], S_0[j_1];$ Swap $S_1[i_2], S_1[j_2];$	
Cycle 3	$i_3 = i_2 + 1;$ $j_3 = j_2 + S_2[i_3];$ $i_4 = i_3 + 1;$ $j_4 = j_3 + S_3[i_4];$		$Z_1 = S_1[S_1[i_1] + S_1[j_1]]$ $Z_2 = S_2[S_2[i_2] + S_2[j_2]]$
Cycle 4		Swap $S_2[i_3], S_2[j_3];$ Swap $S_3[i_4], S_3[j_4];$	
Cycle 5			$Z_3 = S_3[S_3[i_3] + S_3[j_3]]$ $Z_4 = S_4[S_4[i_4] + S_4[j_4]]$

Design 1.5 – Simple hardware pipeline

	Stage 1	Stage 2
Cycle 1	$i_1 = i_0 + 1;$ $j_1 = j_0 + S_0[i_1];$ Swap $S_0[i_1]$, $S_0[j_1]$	
Cycle 2	$i_2 = i_1 + 1;$ $j_2 = j_1 + S_1[i_2];$ Swap $S_1[i_2]$, $S_1[j_2];$	$Z_1 = S_1[S_0[i_1] + S_0[j_1]]$
Cycle 3		$Z_2 = S_2[S_1[i_2] + S_1[j_2]]$

Design 1.5 – Simple hardware pipeline

	Stage 1	Stage 2
Cycle 1	$i_1 = i_0 + 1;$ $j_1 = j_0 + S_0[i_1];$ Swap $S_0[i_1]$, $S_0[j_1]$	
Cycle 2	$i_2 = i_1 + 1;$ $j_2 = j_1 + S_1[i_2];$ Swap $S_1[i_2]$, $S_1[j_2]$;	$Z_1 = S_1[S_0[i_1] + S_0[j_1]]$
Cycle 3		$Z_2 = S_2[S_1[i_2] + S_1[j_2]]$

This approach is independent of the loop unrolling.

Is it possible to merge the two approaches?

Design 2 – Hybrid approach

	Stage 1	Stage 2

Cycle 1	$i_1 = i_0 + 1;$
	$j_1 = j_0 + S_0[i_1];$
	$i_2 = i_1 + 1;$
	$j_2 = j_1 + S_1[i_2];$
	Swap $S_0[i_1]$, $S_0[j_1]$;
	Swap $S_1[i_2]$, $S_1[j_2]$;	

Cycle 2	$i_3 = i_2 + 1;$
	$j_3 = j_2 + S_2[i_3];$
	$i_4 = i_3 + 1;$
	$j_4 = j_3 + S_3[i_4];$
	Swap $S_2[i_3]$, $S_2[j_3]$;
	Swap $S_3[i_4]$, $S_3[j_4]$;
		$Z_1 = S_1[S_1[i_1] + S_1[j_1]]$
		$Z_2 = S_2[S_2[i_2] + S_2[j_2]]$

Cycle 3		$Z_3 = S_3[S_3[i_3] + S_3[j_3]]$ $Z_4 = S_4[S_4[i_4] + S_4[j_4]]$

Design 2 – Hybrid approach

"Two Bytes per Clock throughput for RC4 PRGA"

$$1 \longrightarrow 0.5$$

- $2N$ bytes of output in $N+1$ clock cycles
- Completion of RC4 KSA in 129 clock cycles
- Asymptotically *'two bytes per clock cycle'*

Contributions in RC4 Implementation

Improved the throughputs of common RC4 designs in literature.
Matched the best throughput 1-byte-per-cycle from industry patents.
Provided the best throughput 2-bytes-per-cycle design for RC4.

Year	Result in RC4 implementation	Ref.
2003	3 cycles-per-byte design based on custom pipeline	Kitsos
2003	3 cycles-per-byte design based on multi-port memory	Matthews
2008	1 cycle-per-byte design based on hardware pipelining	Matthews
2010	1 byte-per-cycle design based on loop unrolling	SSMS
2013	2 bytes-per-cycle design based on hardware pipelining combined with loop unrolling in a hybrid model	SCSMS

Open problems in RC4

Open problems in RC4

Key collisions

- Theoretical construction of short colliding key-pairs.
- Search for collision with 16-byte key-pairs in RC4.

Open problems in RC4

Key collisions

- Theoretical construction of short colliding key-pairs.
- Search for collision with 16-byte key-pairs in RC4.

Key recovery

- Narrow the gap of theory and practice in terms of key recovery attacks on WEP and WPA.

Open problems in RC4

Key collisions

- Theoretical construction of short colliding key-pairs.
- Search for collision with 16-byte key-pairs in RC4.

Key recovery

- Narrow the gap of theory and practice in terms of key recovery attacks on WEP and WPA.

Anomaly pairs

- Characterization of all anomalies in RC4.
- Identify and prove all anomaly-dependent biases.

Open problems in RC4

State recovery

- Analysis and improvement of existing results in state recovery.

Open problems in RC4

State recovery

- Analysis and improvement of existing results in state recovery.

Short cycles

- Find lower bound on the length of ‘possible’ cycles in RC4.
- Explicitly find a short cycle in RC4 cipher evolution.

Open problems in RC4

State recovery

- Analysis and improvement of existing results in state recovery.

Short cycles

- Find lower bound on the length of 'possible' cycles in RC4.
- Explicitly find a short cycle in RC4 cipher evolution.

Keystream biases

- Search for all significant biases of the form $(Z_r \star Z_{r+x} = v)$.

Open problems in RC4

State recovery

- Analysis and improvement of existing results in state recovery.

Short cycles

- Find lower bound on the length of 'possible' cycles in RC4.
- Explicitly find a short cycle in RC4 cipher evolution.

Keystream biases

- Search for all significant biases of the form $(Z_r \star Z_{r+x} = v)$.

Hardware implementation

- Area optimization by distributing S-array over memory banks.

Publications

Publications from the Thesis

RC4 Analysis

1. Sourav Sen Gupta, Subhamoy Maitra, Goutam Paul, and Santanu Sarkar. (Non–)random sequences from (non–)random permutations – analysis of RC4 stream cipher. Journal of Cryptology, 2013.
2. Santanu Sarkar, Sourav Sen Gupta, Goutam Paul, and Subhamoy Maitra. Proving TLS-attack related open biases of RC4. IACR ePrint, 2013.
3. Subhamoy Maitra and Sourav Sen Gupta. New long-term glimpse of RC4 stream cipher. In ICISS. Springer LNCS, 2013.
4. Subhamoy Maitra, Goutam Paul, and Sourav Sen Gupta. Attack on broadcast RC4 revisited. In FSE. Springer LNCS, 2011.
5. Sourav Sen Gupta, Subhamoy Maitra, Goutam Paul, and Santanu Sarkar. Proof of empirical RC4 biases and new key correlations. In Selected Areas in Cryptography. Springer LNCS, 2011.

Publications from the Thesis

RC4 Implementation

1. Sourav Sen Gupta, Anupam Chattopadhyay, Koushik Sinha, Subhamoy Maitra, and Bhabani P. Sinha. High-performance hardware implementation for RC4 stream cipher. IEEE Transactions on Computers, 2013.
2. Sourav Sen Gupta, Koushik Sinha, Subhamoy Maitra, and Bhabani P. Sinha. One byte per clock: A novel RC4 hardware. In INDOCRYPT. Springer LNCS, 2010.

Total: 2 journal papers, 4 conference papers, 1 ePrint report.

THANK YOU

for your kind attention