RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

NOW MATTERS

SESSION ID: CRYP-R02

MIXCOLUMNS PROPERTIES AND ATTACKS ON (ROUND-REDUCED) AES WITH A SINGLE SECRET S-BOX

Lorenzo Grassi

Ph.D. Student
IAIK, Graz

MixColumns Properties and Attacks on (round-reduced) AES with a Single Secret S-Box

Lorenzo Grassi

April, 2018

Introduction

A **key-recovery attack** is any adversary's attempt to recover the cryptographic key of an encryption scheme.

Kerckhoffs Principle: the security of a cryptosystem must lie in the choice of its keys only. Everything else should be considered public knowledge.

What happens if part of the crypto-system is instead kept secret?

Introduction

A **key-recovery attack** is any adversary's attempt to recover the cryptographic key of an encryption scheme.

Kerckhoffs Principle: the security of a cryptosystem must lie in the choice of its keys only. Everything else should be considered public knowledge.

What happens if part of the crypto-system is instead kept secret?

Table of Contents

Part I

AES

AES

High-level description of AES:

- block cipher based on a design principle known as *substitution-permutation network*;
- block size of 128 bits = 16 bytes, organized in a 4×4 matrix;
- key size of 128/192/256 bits;
- 10/12/14 rounds:

$$R^i(x) = k^i \oplus MC \circ SR \circ \text{S-Box}(x).$$

SubBytes

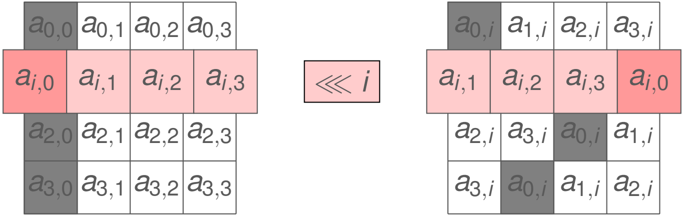
$$\begin{array}{|c|c|c|c|}\hline a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ \hline a_{1,0} & a_{1,1} & \boxed{a_{i,j}} & a_{1,3} \\ \hline a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ \hline a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \\ \hline \end{array} \quad \boxed{\text{S-Box}} \quad \begin{array}{|c|c|c|c|}\hline b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ \hline b_{1,0} & b_{1,1} & \boxed{b_{i,j}} & b_{1,3} \\ \hline b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ \hline b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \\ \hline \end{array}$$

- Bytes are transformed by invertible S-Box with

$$b_{i,j} = \text{S-Box}(a_{i,j})$$

- Same S-Box (lookup table) for the whole cipher:
  - based on multiplicative inverse in $GF(2^8)$

ShiftRows



- Rows are rotated over 4 different offsets
- "*Optimal Diffusion*": two bytes in the same column are mapped into different columns after ShiftRows operation

MixColumns

$$
\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,j} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,j} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,j} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,j} & a_{3,3} \end{bmatrix}
\quad \times \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \quad
\begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,j} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,j} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,j} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,j} & b_{3,3} \end{bmatrix}
$$

- Columns transformed by 4×4 matrix over $GF(2^8)$
- MDS matrix (Branch number $= 5$)
- Together with ShiftRows, *high diffusion* over multiple rounds

AES with a single Secret S-Box

Consider AES with a single secret S-Box: the size of the secret information increases from 128-256 bits to

$$128 + \log_2 2^8! = 1812$$
$$256 + \log_2 2^8! = 1940$$

How does the security of the AES change when the S-Box is replaced by a secret S-Box, about which the adversary has no knowledge?

Part II

AES with a single Secret S-Box - State of the Art

AES with a single Secret S-Box - 1*st* Strategy

A possible strategy exploited by many attacks ([BS01], [TKK+15], ...) in the literature:

1. determine the secret S-Box up to additive constants, i.e.

$$\text{S-Box}(a \oplus x) \oplus b;$$

2. exploit this knowledge to find the key (e.g. using an integral attack).

AES with a single Secret S-Box - 2*nd* Strategy

It is also possible to find directly the key, i.e. without finding or exploiting any information of the S-Box!

Exploit the fact that each row of the MixColumns matrix

$$MC \equiv \begin{bmatrix} 0x02 & 0x03 & 0x01 & 0x01 \\ 0x01 & 0x02 & 0x03 & 0x01 \\ 0x01 & 0x01 & 0x02 & 0x03 \\ 0x03 & 0x01 & 0x01 & 0x02 \end{bmatrix}$$

has two identical elements for each row!

Idea of the Attack

Guess one byte of the key δ and consider the set V_δ

$$V_\delta = \{(p^i, c^i) \quad \forall i = 0, ..., 2^8 - 1 \mid p^i_{0,0} \oplus p^i_{1,1} = \delta$$
$$\text{and} \quad p^i_{k,l} = p^j_{k,l} \quad \forall (k,l) \neq \{(0,0),(1,1)\} \text{ and } \forall i \neq j\}.$$

Since $MC_{2,0} = MC_{2,1}$:

- If $\delta = k_{0,0} \oplus k_{1,1}$, given $p^1, p^2 \in V_\delta$ then $R(p^1)_{2,0} = R(p^2)_{2,0}$ with prob. 1;
- If $\delta \neq k_{0,0} \oplus k_{1,1}$, given $p^1, p^2 \in V_\delta$ then $R(p^1)_{2,0} = R(p^2)_{2,0}$ with prob. 2^{-8}.

Idea of the Attack

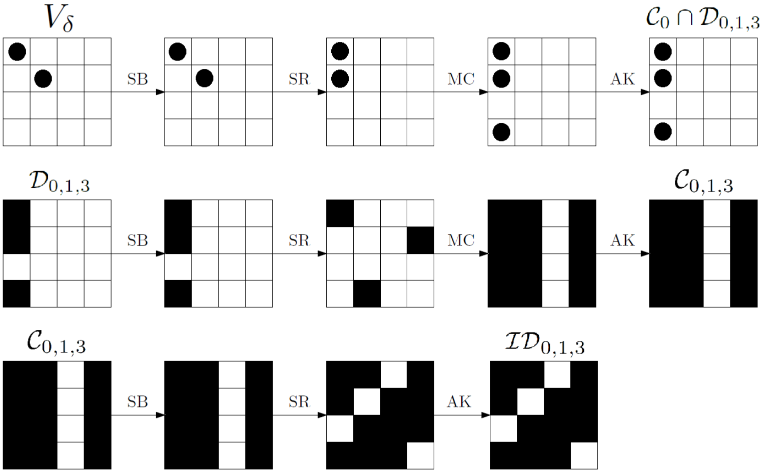
Guess one byte of the key δ and consider the set V_δ

$$V_\delta = \{(p^i, c^i) \quad \forall i = 0, ..., 2^8 - 1 \,|\, p^i_{0,0} \oplus p^i_{1,1} = \delta \\ \text{and} \quad p^i_{k,l} = p^j_{k,l} \quad \forall (k,l) \neq \{(0,0),(1,1)\} \text{ and } \forall i \neq j\}.$$

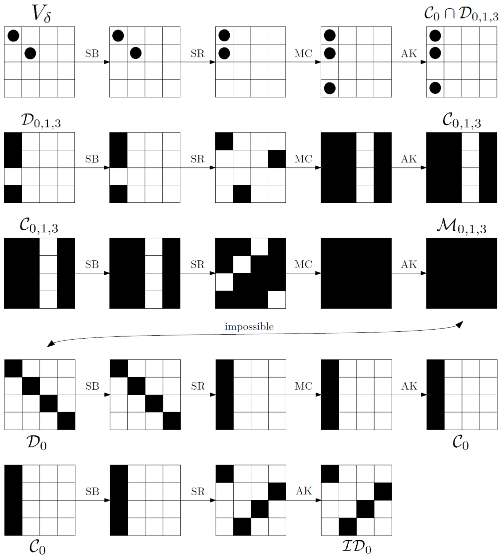
Since $MC_{2,0} = MC_{2,1}$:

- If $\delta = k_{0,0} \oplus k_{1,1}$, given $p^1, p^2 \in V_\delta$ then $R(p^1)_{2,0} = R(p^2)_{2,0}$ with prob. 1;
- If $\delta \neq k_{0,0} \oplus k_{1,1}$, given $p^1, p^2 \in V_\delta$ then $R(p^1)_{2,0} = R(p^2)_{2,0}$ with prob. 2^{-8}.

Key-Recovery Attack on 3-round AES



Key-Recovery Attack on 5-round AES



Part III

AES with a single Secret S-Box - Multiple-of-n Property

Multiple-of-8 Property - [GRR17]

Consider a set of 2^{32} chosen plaintexts with one active diagonal

$$\begin{bmatrix} A & C & C & C \\ C & A & C & C \\ C & C & A & C \\ C & C & C & A \end{bmatrix}$$

and the corresponding ciphertexts after 5-round AES.

The number N of different pairs of ciphertexts (c^1, c^2) that are equal in one fixed anti-diagonal (final MC omitted), e.g.

$$c^1 \oplus c^2 = \begin{bmatrix} ? & ? & ? & 0 \\ ? & ? & 0 & ? \\ ? & 0 & ? & ? \\ 0 & ? & ? & ? \end{bmatrix}$$

is always a multiple of 8 with prob. 1 *independently of the secret key, of the details of the S-Box and of the MixColumns matrix.*

Multiple-of-n Property - 5-round AES

Guess one byte of the key δ and consider the set of 2^{40} plaintexts V_δ

$$V_\delta \equiv \left\{ a \oplus \begin{bmatrix} x_0 & y & 0 & 0 \\ 0 & x_1 & y \oplus \delta & 0 \\ 0 & 0 & x_2 & 0 \\ 0 & 0 & 0 & x_3 \end{bmatrix} \middle| \forall x_0, \ldots, x_3, y \in \mathbb{F}_{2^8} \right\}$$

Let N the number of different pairs of ciphertexts (c^1, c^2) that are equal in one fixed anti-diagonal, e.g.

$$c^1 \oplus c^2 = \begin{bmatrix} ? & ? & ? & 0 \\ ? & ? & 0 & ? \\ ? & 0 & ? & ? \\ 0 & ? & ? & ? \end{bmatrix}$$

(final MC omitted for simplicity)

Multiple-of-n Property - 5-round AES

Guess one byte of the key δ and consider the set of 2^{40} plaintexts V_δ

$$V_\delta \equiv \left\{ a \oplus \begin{bmatrix} x_0 & y & 0 & 0 \\ 0 & x_1 & y \oplus \delta & 0 \\ 0 & 0 & x_2 & 0 \\ 0 & 0 & 0 & x_3 \end{bmatrix} \middle| \, \forall x_0, \ldots, x_3, y \in \mathbb{F}_{2^8} \right\}$$

Let N the number of different pairs of ciphertexts (c^1, c^2) that are equal in one fixed anti-diagonal, e.g.

$$c^1 \oplus c^2 = \begin{bmatrix} ? & ? & ? & 0 \\ ? & ? & 0 & ? \\ ? & 0 & ? & ? \\ 0 & ? & ? & ? \end{bmatrix}$$

(final MC omitted for simplicity)

Multiple-of-n Property - 5-round AES

Let N the number of different pairs of ciphertexts (c^1, c^2) that are equal in one fixed anti-diagonal (final MC omitted for simplicity), i.e. that belong to the same coset of a particular subspace $\mathcal{M}$.

Since $MC_{3,0} = MC_{3,1}$:

- If $\delta = k_{0,1} \oplus k_{1,2}$, N is a multiple of 2 - i.e. $N = 2 \cdot N'$ - with prob. 1;
- If $\delta \neq k_{0,1} \oplus k_{1,2}$, N is a multiple of 2 with prob. 50% (same probability to be even or odd).

Sketch of the Proof (1/2)

If $\delta = k_{0,1} \oplus k_{1,2}$

$$R(V_\delta) \equiv \left\{ b \oplus \begin{bmatrix} x_0 & y & 0 & 0 \\ x_1 & 0x03 \cdot y & 0 & 0 \\ x_2 & 0 & 0 & 0 \\ x_3 & 0x02 \cdot y & 0 & 0 \end{bmatrix} \middle| \forall x_0, \ldots, x_3, y \in \mathbb{F}_{2^8} \right\}$$

independently of the secret S-Box.

Given $p^1 \equiv \langle x_0, x_1, x_2, x_3, y \rangle$ and $p^2 \equiv \langle \tilde{x}_0, \tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{y} \rangle$ in $R(V_\delta)$, consider the following two cases:

- $x_1 \neq \tilde{x}_1$
- $x_1 = \tilde{x}_1$

Sketch of the Proof (1/2)

If $\delta = k_{0,1} \oplus k_{1,2}$

$$R(V_\delta) \equiv \left\{ b \oplus \begin{bmatrix} x_0 & y & 0 & 0 \\ x_1 & \texttt{0x03} \cdot y & 0 & 0 \\ x_2 & 0 & 0 & 0 \\ x_3 & \texttt{0x02} \cdot y & 0 & 0 \end{bmatrix} \middle| \forall x_0, \ldots, x_3, y \in \mathbb{F}_{2^8} \right\}$$

independently of the secret S-Box.

Given $p^1 \equiv \langle x_0, x_1, x_2, x_3, y \rangle$ and $p^2 \equiv \langle \tilde{x}_0, \tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{y} \rangle$ in $R(V_\delta)$, consider the following two cases:

- $x_1 \neq \tilde{x}_1$
- $x_1 = \tilde{x}_1$

Sketch of the Proof (2/2)

Given $p^1 \equiv \langle x_0, x_1, x_2, x_3, y \rangle$ and $p^2 \equiv \langle \tilde{x}_0, \tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{y} \rangle$ in $R(V_\delta)$.

- If $x_1 \neq \tilde{x}_1$, it is possible to prove that

$$R^4(p^1) \oplus R^4(p^2) \in \mathcal{M} \qquad \text{iff} \qquad R^4(q^1) \oplus R^4(q^2) \in \mathcal{M}$$

  where $q^1 \equiv \langle x_0, \tilde{\mathbf{x}}_\mathbf{1}, x_2, x_3, y \rangle$ and $q^2 \equiv \langle \tilde{x}_0, \mathbf{x}_\mathbf{1}, \tilde{x}_2, \tilde{x}_3, \tilde{y} \rangle$ in $R(V_\delta)$.

- If $x_1 = \tilde{x}_1$, it is possible to prove that

$$R^4(p^1) \oplus R^4(p^2) \in \mathcal{M} \qquad \text{iff} \qquad R^4(q^1) \oplus R^4(q^2) \in \mathcal{M}$$

  where $q^1 \equiv \langle x_0, \mathbf{w}, x_2, x_3, y \rangle$ and $q^2 \equiv \langle \tilde{x}_0, \mathbf{w}, \tilde{x}_2, \tilde{x}_3, \tilde{y} \rangle$ in $R(V_\delta)$ **for all** $w \in \mathbb{F}_{2^8}$.

Sketch of the Proof (2/2)

Given $p^1 \equiv \langle x_0, x_1, x_2, x_3, y\rangle$ and $p^2 \equiv \langle \tilde{x}_0, \tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{y}\rangle$ in $R(V_\delta)$.

- If $x_1 \neq \tilde{x}_1$, it is possible to prove that

  $$R^4(p^1) \oplus R^4(p^2) \in \mathcal{M} \qquad \text{iff} \qquad R^4(q^1) \oplus R^4(q^2) \in \mathcal{M}$$

  where $q^1 \equiv \langle x_0, \tilde{\mathbf{x}}_\mathbf{1}, x_2, x_3, y\rangle$ and $q^2 \equiv \langle \tilde{x}_0, \mathbf{x}_\mathbf{1}, \tilde{x}_2, \tilde{x}_3, \tilde{y}\rangle$ in $R(V_\delta)$.

- If $x_1 = \tilde{x}_1$, it is possible to prove that

  $$R^4(p^1) \oplus R^4(p^2) \in \mathcal{M} \qquad \text{iff} \qquad R^4(q^1) \oplus R^4(q^2) \in \mathcal{M}$$

  where $q^1 \equiv \langle x_0, \mathbf{w}, x_2, x_3, y\rangle$ and $q^2 \equiv \langle \tilde{x}_0, \mathbf{w}, \tilde{x}_2, \tilde{x}_3, \tilde{y}\rangle$ in $R(V_\delta)$ **for all** $w \in \mathbb{F}_{2^8}$.

Part IV

AES with a single Secret S-Box - “Weaker” Property of MixColumns Matrix

“Weaker” Property of MixColumns Matrix

Is there any weaker property of the MixColumns matrix that allows to find directly the key, i.e. without finding or exploiting any information of S-Box?

Yes! Exploit the fact that for each row of the MixColumns matrix

$$MC \equiv \begin{bmatrix} \texttt{0x02} & \texttt{0x03} & \texttt{0x01} & \texttt{0x01} \\ \texttt{0x01} & \texttt{0x02} & \texttt{0x03} & \texttt{0x01} \\ \texttt{0x01} & \texttt{0x01} & \texttt{0x02} & \texttt{0x03} \\ \texttt{0x03} & \texttt{0x01} & \texttt{0x01} & \texttt{0x02} \end{bmatrix}$$

the XOR-sum of two or more elements is equal to zero!

Idea of the Attack

Guess two bytes of the key $\delta = (\delta_1, \delta_2)$ and consider the set V_δ

$$V_\delta = \{(p^i, c^i)\, \forall i = 0, ..., 2^8 - 1 \mid p^i_{0,0} \oplus p^i_{1,1} = \delta_1,\ p^i_{0,0} \oplus p^i_{2,2} = \delta_2 \\ \text{and} \quad p^i_{k,l} = p^j_{k,l} \quad \forall (k,l) \neq \{(0,0),(1,1),(2,2)\} \text{ and } \forall i \neq j\}.$$

Since $MC_{0,0} \oplus MC_{0,1} \oplus MC_{0,2} = 0$ and $MC_{1,0} \oplus MC_{1,1} \oplus MC_{1,2} = 0$:

- If $\delta_1 = k_{0,0} \oplus k_{1,1}$ and $\delta_2 = k_{0,0} \oplus k_{2,2}$, given $p^1, p^2 \in V_\delta$ then $R(p^1)_{0,0} = R(p^2)_{0,0}$ and $R(p^1)_{1,0} = R(p^2)_{1,0}$ with prob. 1;
- If $\delta_1 \neq k_{0,0} \oplus k_{1,1}$ and/or $\delta_2 \neq k_{0,0} \oplus k_{2,2}$, given $p^1, p^2 \in V_\delta$ then $R(p^1)_{0,0} = R(p^2)_{0,0}$ and $R(p^1)_{1,0} = R(p^2)_{1,0}$ with prob. 2^{-16}.

Idea of the Attack

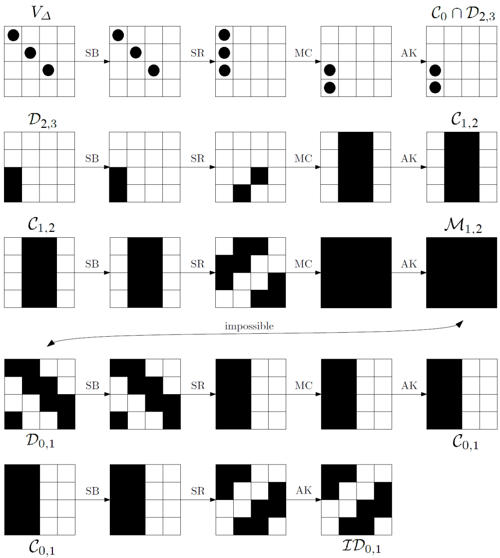
Guess two bytes of the key $\delta = (\delta_1, \delta_2)$ and consider the set V_δ

$$V_\delta = \{(p^i, c^i)\, \forall i = 0, ..., 2^8 - 1 \mid p^i_{0,0} \oplus p^i_{1,1} = \delta_1,\; p^i_{0,0} \oplus p^i_{2,2} = \delta_2 \\ \text{and} \quad p^i_{k,l} = p^j_{k,l} \quad \forall (k,l) \neq \{(0,0),(1,1),(2,2)\} \text{ and } \forall i \neq j\}.$$

Since $MC_{0,0} \oplus MC_{0,1} \oplus MC_{0,2} = 0$ and $MC_{1,0} \oplus MC_{1,1} \oplus MC_{1,2} = 0$:

- If $\delta_1 = k_{0,0} \oplus k_{1,1}$ and $\delta_2 = k_{0,0} \oplus k_{2,2}$, given $p^1, p^2 \in V_\delta$ then $R(p^1)_{0,0} = R(p^2)_{0,0}$ and $R(p^1)_{1,0} = R(p^2)_{1,0}$ with prob. 1;
- If $\delta_1 \neq k_{0,0} \oplus k_{1,1}$ and/or $\delta_2 \neq k_{0,0} \oplus k_{2,2}$, given $p^1, p^2 \in V_\delta$ then $R(p^1)_{0,0} = R(p^2)_{0,0}$ and $R(p^1)_{1,0} = R(p^2)_{1,0}$ with prob. 2^{-16}.

Key-Recovery Attack on 5-round AES



Multiple-of-n Property - 5-round AES

Guess two bytes of the key $\delta = (\delta_1, \delta_2)$ and consider the set of 2^{40} plaintexts V_δ

$$V_\delta \equiv \left\{ a \oplus \begin{bmatrix} x_0 & y & 0 & 0 \\ 0 & x_1 & y \oplus \delta_1 & 0 \\ 0 & 0 & x_2 & y \oplus \delta_2 \\ 0 & 0 & 0 & x_3 \end{bmatrix} \middle| \, \forall x_0, ..., x_3, y \in \mathbb{F}_{2^8} \right\}$$

Let N the number of different pairs of ciphertexts (c^1, c^2) that are equal in one fixed anti-diagonal (final MC omitted). If

$$\delta_1 = k_{0,1} \oplus k_{1,2} \qquad \text{and} \qquad \delta_2 = k_{0,1} \oplus k_{2,3}$$

then N is a multiple of 4 - i.e. $N = 4 \cdot N'$ - with prob. 1.

Multiple-of-n Property - 5-round AES

Guess two bytes of the key $\delta = (\delta_1, \delta_2)$ and consider the set of 2^{40} plaintexts V_δ

$$V_\delta \equiv \left\{ a \oplus \begin{bmatrix} x_0 & y & 0 & 0 \\ 0 & x_1 & y \oplus \delta_1 & 0 \\ 0 & 0 & x_2 & y \oplus \delta_2 \\ 0 & 0 & 0 & x_3 \end{bmatrix} \middle| \, \forall x_0, \ldots, x_3, y \in \mathbb{F}_{2^8} \right\}$$

Let N the number of different pairs of ciphertexts (c^1, c^2) that are equal in one fixed anti-diagonal (final MC omitted). If

$$\delta_1 = k_{0,1} \oplus k_{1,2} \qquad \text{and} \qquad \delta_2 = k_{0,1} \oplus k_{2,3}$$

then N is a multiple of 4 - i.e. $N = 4 \cdot N'$ - with prob. 1.

Number of *Circulant* Matrices

Case: $\mathbb{F}_{2^4}^{4\times 4}$

	Invertible Matrices	MDS Matrices
Total	61 440	16 560
Two Equal Coeff.	32 640 (53.125%)	10 080 (60.87%)
Zero XoR-Sum	45 600 (74.22%)	12 480 (75.36%)

Case: $\mathbb{F}_{2^8}^{4\times 4}$

	Invertible Matrices	MDS Matrices
Total	4 278 190 080	4 015 735 920
Two Equal Coeff.	165 550 080 (3.87%)	126 977 760 (3.16%)
Zero XoR-Sum	293 556 000 (6.87%)	249 418 560 (6.21%)

Our Results

Attack	Rounds	Data	Computation	Memory
I* [TKK+15]	4.5 - 5	2^{40} CC	$2^{38.7}$ E	2^{40}
I* [TKK+15]	4.5 - 5	2^{40} CP	$2^{54.7}$ E	2^{40}
Mult-of-n	**4.5 – 5**	**$2^{53.25}$ CP**	**$2^{52.6}$ E**	**2^{16}**
Mult-of-n	**4.5 – 5**	**$2^{53.6}$ CP**	**$2^{48.96}$ E**	**2^{40}**
ImD	**4.5 – 5**	**$2^{76.3}$ CP**	**$2^{74.9}$ E**	**2^{8}**
ImD [GRR16]	4.5 - 5	2^{102} CP	2^{107} M $\approx 2^{100.4}$ E	2^{8}
I [SLG+16]	5	2^{128} CC	$2^{129.6}$ XOR	small

I: Integral, ImD: Impossible Differential, Mult-of-*n*: Multiple-of-*n*

Symbol *: attack in which one must first find the S-Box (up to additive constants), and exploit this information to find the key

Part V

Open Problems

Future Works

Cryptanalysis for the case of AES with a single secret S-Box.

- Look for *weaker properties of the Linear Layer* that allows to set up a key-recovery attack in the case of secret S-Box
- What if ***all*** *the S-Box are different and still secret*?
- Until now we have considered the case of secret S-Box and known Linear Layer. What happens in the opposite situation of secret Linear Layer and known S-Box?

Thanks for your attention!

Questions?

Comments?

References I

A. Biryukov and A. Shamir,
Structural Cryptanalysis of SASAS
EUROCRYPT 2001

J. Daemen, L. Knudsen and V. Rijmen,
The Block Cipher Square
FSE 1997

L. Grassi, C. Rechberger and S. Rønjom,
Subspace Trail Cryptanalysis and its Applications to AES
IACR Transactions on Symmetric Cryptology 2016

References II

L. Grassi, C. Rechberger and S. Rønjom,
A New Structural-Differential Property of 5-Round AES
EUROCRYPT 2017

B. Sun and M. Liu and J. Gou and L. Qu and V. Rijmen,
New Insights on AES-Like SPN Ciphers
CRYPTO 2016

T. Tiessen, L.R. Knudsen, S. Kölbl and M.M. Lauridsen,
Security of the AES with a Secret S-Box
FSE 2015

RSAConference2018

San Francisco | April 16 – 20 | Moscone Center

SESSION ID: CRYP-R02

COUNT-THEN-PERMUTE: A PRECISION-FREE ALTERNATIVE TO INVERSION SAMPLING

Kentarou Sasaki

Researcher
NEC Corporation
@sierrarries

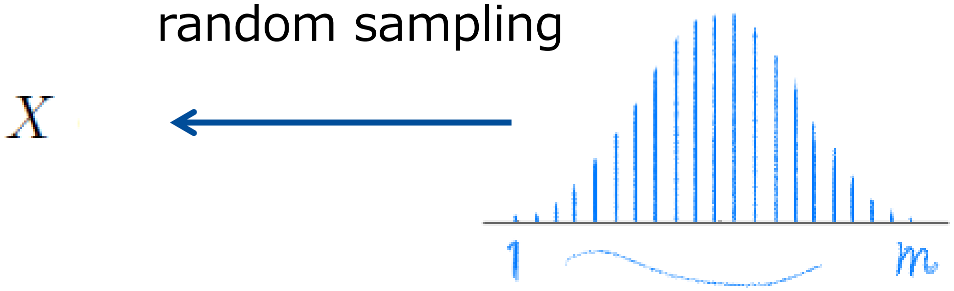
Count-then-Permute: a Precision-free Alternative to Inversion Sampling

CT-RSA 2018
San Francisco, Apr 19, 2018

Kazuhiko Minematsu
Kentarou Sasaki
Yuki Tanaka
(NEC Corporation)

\Orchestrating a brighter world NEC

Sampling from a discrete distribution



Settings

X : random variable whose value is in $\{1, 2, \ldots, m\}$

$p(X)$: distribution of X

$p_i = \Pr[X = i] \ (1 \leq i \leq m)$

k-bit: precision of p_i

Inversion Sampler

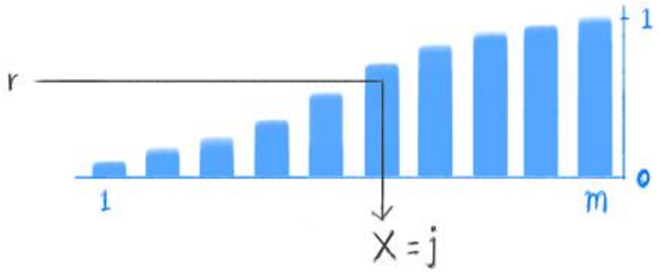
- Classical generic sampler, simple and easy to implement (see e.g. Debroye's book [Dev86])
- Fast, if memory access is fast
- Create Cumulative Distribution Function (CDF) table in advance and sample with it

CDF table

- table = $[s_1, s_2, \ldots, s_m]$ $\quad s_i = \Pr[1 \leq X \leq i] = \sum_{j=1}^{i} p_j$

Algorithm

1. $r \leftarrow$ uniform distribution on interval $[0, 1]$
2. return $\min\{j \mid r \leq s_j\}$



Inversion Sampler

- Require $O(km)$ memory size
  - Table = $[s_1, s_2, \ldots, s_m]$
  - Each s_i is a k-bit floating point number

- Physical uniform random number generator is quite costly in general

- Instead, symmetric key cryptography such as block cipher is used as pseudorandom number generator in practice

Motivation

Problems

- Precision k needs to be very high (e.g. 128 or 256) for cryptographic usage
- For example, Discrete Gaussian sampling in lattice cryptography
- Precision affects security level
- Table size also affects sampling speed
  - Smaller table may fit into cache

Our Goal: a generic sampler with precision- independent table size

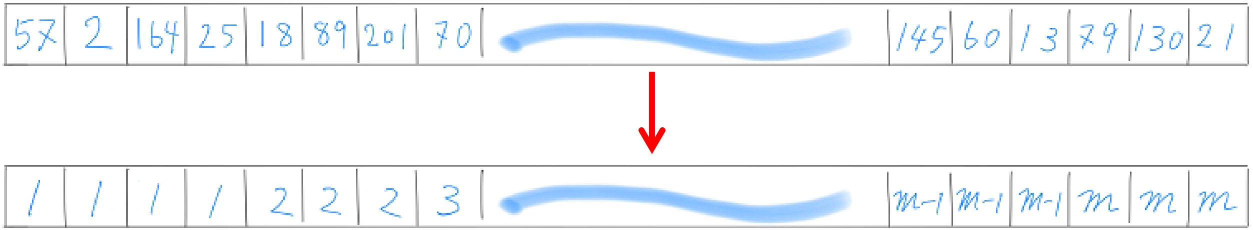
Naïve idea

- Let N be the number of sample we need
- First, sample all N samples and sort them

- Then apply a random permutation to the sorted N samples and output from the first

Naïve idea

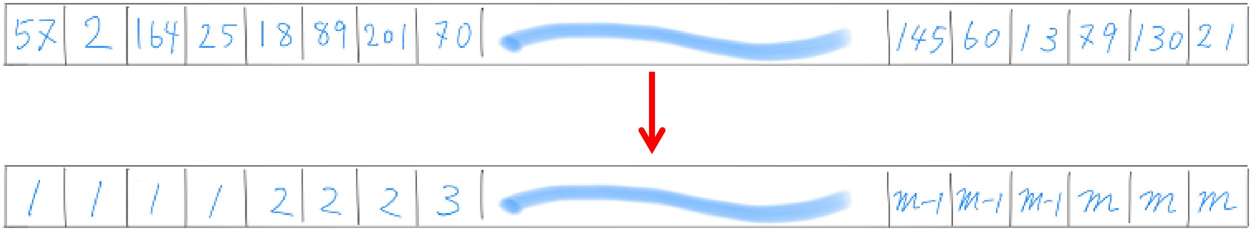
- Let N be the number of sample we need
- First, sample all N samples and sort them



- Then apply a random permutation to the sorted N samples and output from the first

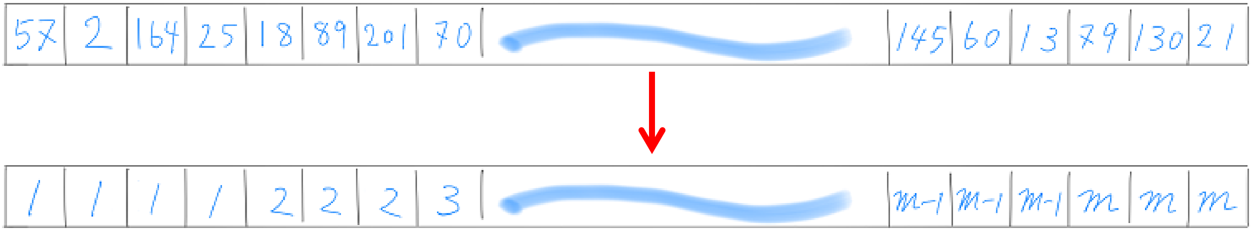
Naïve idea

- Let N be the number of sample we need
- First, sample all N samples and sort them



- Then apply a random permutation to the sorted N samples and output from the first

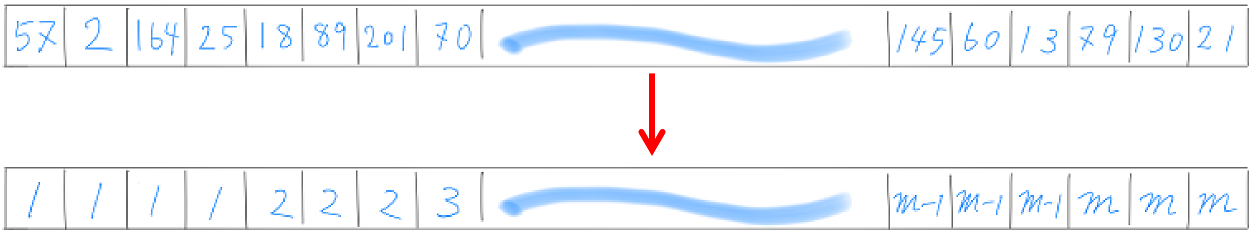
- Let N be the number of sample we need
- First, sample all N samples and sort them



- Then apply a random permutation to the sorted N samples and output from the first

Naïve idea

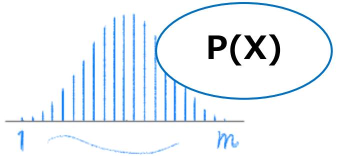
Precomputation



On-line Sampling

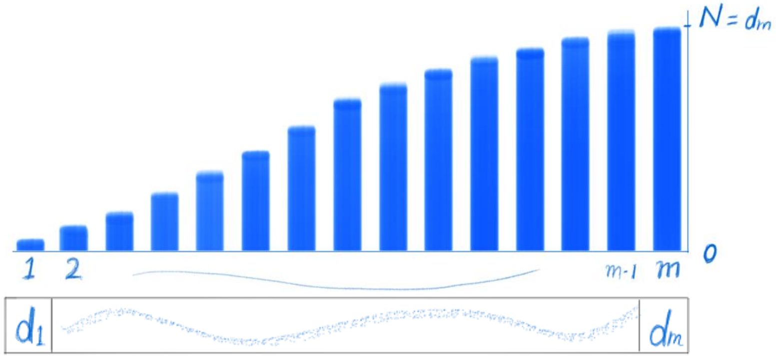
Naïve idea: Precomputation

- Sample N samples from the distribution $p(X)$ of X



↓ N samples

- Create a cumulative histogram of them

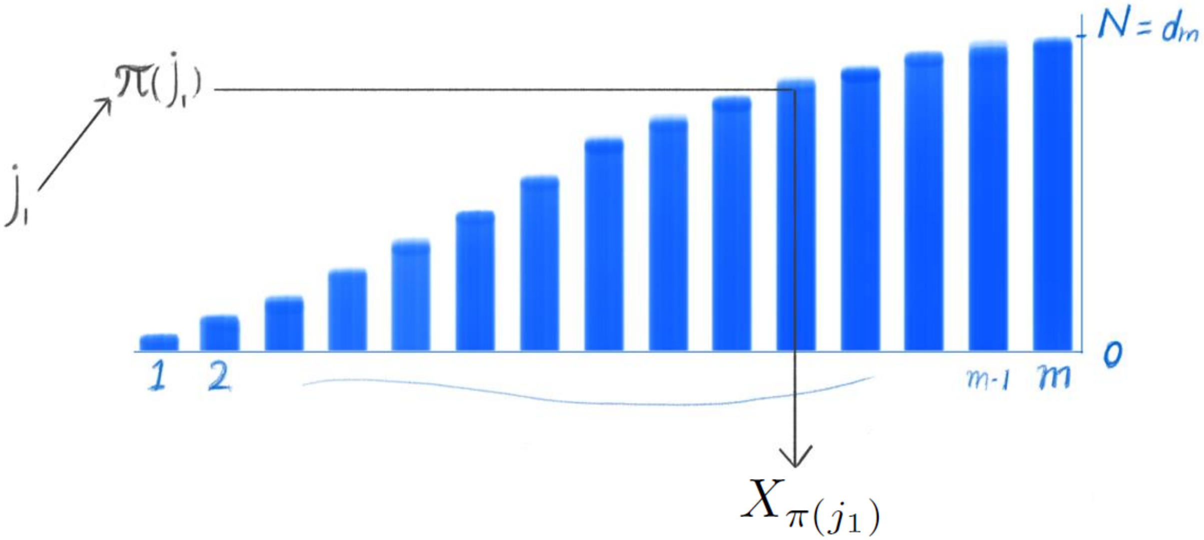


$$d_i = \#\{j \in \{1, 2, \dots, N\} | X_j <= i\}$$

Naïve idea: On-line sampling

Perform random sampling without replacement from the sorted N samples

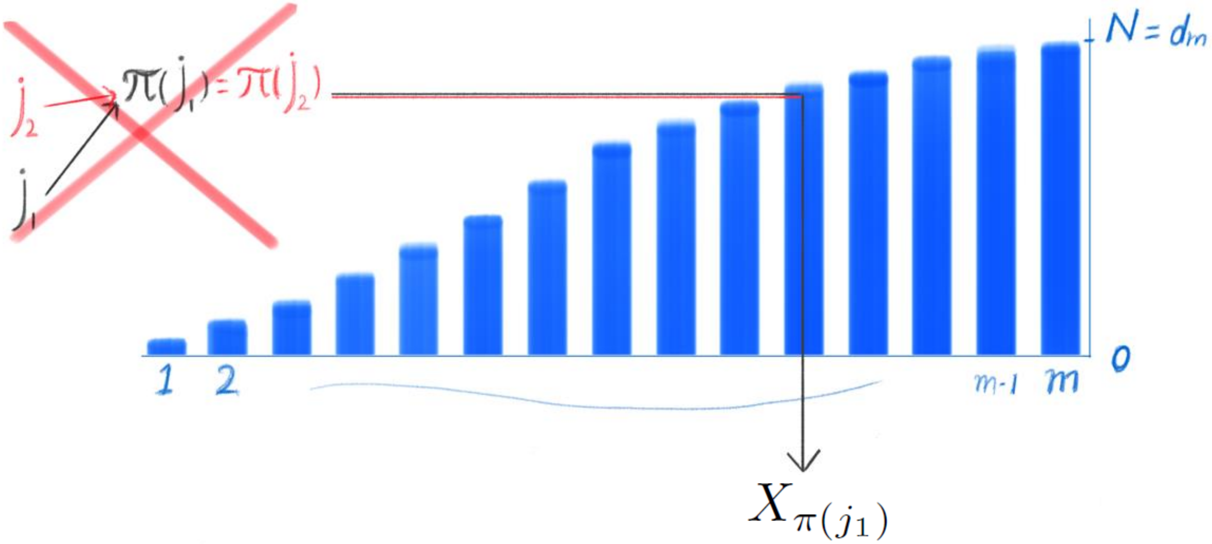
- Let $\pi(\cdot)$ be a random permutation on $\{1, 2, \ldots, N\}$
- j-th sample is a $\pi(j)$-th sample on the table



Naïve idea: On-line sampling

Perform random sampling without replacement from the sorted N samples

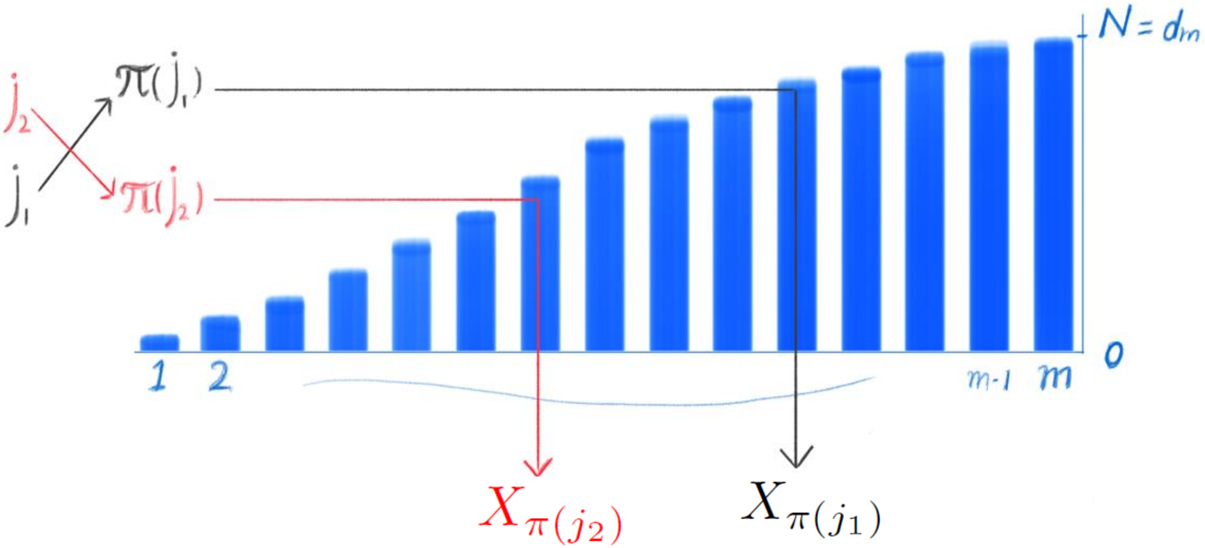
- Let $\pi(\cdot)$ be a random permutation on $\{1, 2, \ldots, N\}$
- For different j, $\pi(j)$ is different



Naïve idea: On-line sampling

Perform random sampling without replacement from the sorted N samples

- Let $\pi(\cdot)$ be a random permutation on $\{1, 2, \ldots, N\}$
- For different j, $\pi(j)$ is different



(Naïve form of) Count-then-Permute (CP) Sampler

Let $N = 2^n$

Let $\pi(\cdot)$ be a random permutation over $\{1, 2, \ldots, N\}$

- Precomputation
  1. Sample N samples
  2. Create a cumulative histogram of the samples $(X_1, \ldots, X_N)$

     table = $[d_1, d_2, \ldots, d_m]$ $\quad d_i = \#\{j \in \{1, 2, \ldots, N\} | X_j <= i\}$

- Online sampling

1. For $1 \leq j \leq N$
   1. $r \longleftarrow \pi(j)$
   2. Return $\min\{i \mid r \leq s_i\}$

Table size is $O(mn)$ and independent of precision k

Hence memory is independent of k and smaller if $n < k$

Two difficulties

Precomputation is totally pointless

- Sampling all N samples: exactly the original problem ☹

Random permutation on $\{1, 2, \ldots, N\}$ is infeasible when N is large

- $O(N)$ time e.g. by Knuth shuffle

Two difficulties

Precomputation is totally pointless

- Sampling all N samples: exactly the original problem ☹

-> Directly sample a cumulative histogram

Random permutation on $\{1, 2, \ldots, N\}$ is infeasible when N is large

- $O(N)$ time e.g. by Knuth shuffle

-> Employ computationally secure block cipher as pseudorandom permutation

Two difficulties

Precomputation is totally pointless

- Sampling all N samples: exactly the original problem ☹

-> Directly sample a cumulative histogram

Random permutation on $\{1, 2, \ldots, N\}$ is infeasible when N is large

- $O(N)$ time e.g. by Knuth shuffle

-> Employ computationally secure block cipher as pseudorandom permutation

Block cipher

A secure block cipher = pseudorandom permutation

cannot be distinguished by random permutation by any polynomial-time adversary

Parameters: block size and key length

- Block size n $\longleftrightarrow$ permutation over $\{1, 2, \ldots, N\}$
- Key length $\longleftrightarrow$ security level

Examples

- Block size 128: AES
- Block size 64: lightweight block cipher such as PRESENT [BKLPPRSV07]

On-line sampling with Block cipher

- Random permutation can be replaced by a block cipher E with appropriate key length
- Correctness of the online sampling is up to the pseudo randomness of E

- Algorithm

  Let $N = 2^n$, Let E_K be a block cipher of block size n with key K
- Precomputation

1. Sample N samples $(X_1, \ldots, X_N)$
2. Sort, count and create a histogram of the samples

   table = $[d_1, d_2, \ldots, d_m]$
3. $K \longleftarrow$ Key space #Sampling of block cipher key

- Online sampling

1. For $1 \leq j \leq N$
   1. $r \longleftarrow E_K(j)$
   2. Return $\min\{i \mid r \leq s_i\}$

Two difficulties

Precomputation is totally pointless

- Sampling all N samples: exactly the original problem ☹

-> Directly sample a cumulative histogram

Random permutation on $\{1, 2, \ldots, N\}$ is infeasible when N is large

- $O(N)$ time e.g. by Knu… s…uffl

SOLVED

-> Employ computationally secure block cipher as pseudorandom permutation

Two difficulties

Precomputation is totally pointless

- Sampling all N samples: exactly the original problem ☹

-> Directly sample a cumulative histogram

Random permutation on $\{1, 2, \ldots, N\}$ is infeasible when N is large

- $O(N)$ time e.g. by Kn[illegible] s[illegible]uffl[illegible]

SOLVED

-> Employ computationally secure block cipher as pseudorandom permutation

Precomputation

- Sampling directly a cumulative histogram is reduced to iterative sampling from binomial distributions
  - Cumulative histogram: $[d_1, d_2, \ldots, d_m]$
  - Histogram: $[c_1, c_2, \ldots, c_m]$ $\quad c_i = \#\{j \in \{1, 2, \ldots, N\} | X_j = i\} \quad d_i = \sum_{j=1}^{i} c_j$
  - Probability of a histogram to be $[c_1, c_2, \ldots, c_m]$:

$$Pr(\text{histogram} = (c_1, \cdots, c_m)) = \frac{N!}{c_1! \cdot c_2! \cdots c_m!} \cdot p_1^{c_1} p_2^{c_2} \cdots p_m^{c_m}$$

  - Conditional probability that i -th bin is c_i given bins $(c_1, \ldots, c_{i-1})$

$$Pr(i\text{-th bin is } c_i | c_1, c_2, \ldots, c_{i-1}) = \mathcal{B}\left(N - c_1 - \cdots - c_{i-1}, \frac{p_i}{1 - \sum_{j=1}^{i} p_j}\right)$$

$\mathcal{B}(N, p)$: binomial distribution

- Popular samplers for $\mathcal{B}(N, p)$ require $O(N)$ time

- Bringmann et.al. [BKP14] and Farach-Colten and Tsai [FT15] showed that exact sampling from binomial distribution is possible in expected or with high probability $O(\log N)$ time

CP Sampler Algorithm: Precomputation

Let $N = 2^n$

Let E_K be a block cipher

$c_0, d_0 \leftarrow 0, d_m \leftarrow N, p'_1 = p_1$

for $i = 0$ to $m - 1$ **do**

  $c_i \leftarrow \mathcal{B}(N - d_{i-1}, p'_i)$ #Binomial distribution sampling

  $d_i \leftarrow d_{i-1} + c_i$ #i-th bin of cumulative histogram

  $p'_{i+1} \leftarrow \frac{p_{i+1}}{1 - \sum_{j=1}^{i} p_j}$

end for

Table $\leftarrow (d_1, \ldots, d_m)$

$K \leftarrow \mathcal{K}$: Key space #Sampling a block cipher key

return Table, K

Two difficulties

Precomputation is totally pointless

- Sampling all N samples: exactly the original problem ☹

-> Directly sample a cumulative histogram

SOLVED

Random permutation on $\{1, 2, \ldots, N\}$ is infeasible when N is large

- $O(N)$ time e.g. by Knuth shuffle

-> Employ computationally secure block cipher as pseudorandom permutation

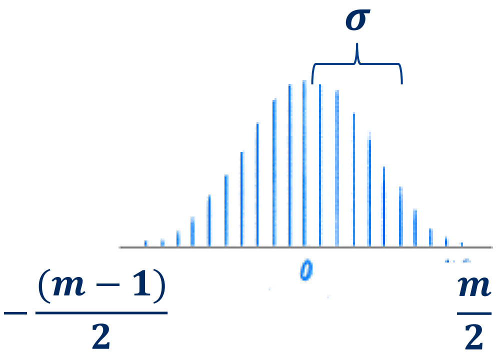
SOLVED

Experimental implementation of CP sampler

- To get an initial idea on the performance of CP Sampler in comparison to inversion sampler

Target distribution: Discrete Gaussian

- Parameters taken from several lattice cryptographic schemes [Micc11][BG14][Lyu12]
  - the bottle neck of speed is often the underlying discrete Gaussian sampling



σ : standard deviation

$S = \sqrt{2\pi}\sigma$

Implementation details

Baseline: Inversion Sampler (IS)

- For floating point calculation we used GMP MPFR library
- For random generator we employed Mersenne Twister (default of GMP rand function)

Block cipher in CP Sampler: AES in C and AESNI

- AES128: block size 128, key length 128
- AES256: block size 128, key length 256

Remarks

- Precomputation is not implemented. Instead of a histogram, we used the table of the expected numbers of samples
- Binary search is implemented for both CP and IS

Results

Speed is an average of 100,000 samples

Scheme(S, m)	prec.	Inversion		Count-then-Permute		
		speed	memory	speed C	speed NI	memory
BG(145, 1624)	128	437	25.4	480	351	25.4
BG(561, 6272)	128	478	245.6	553	406	245.6
Lyu(6737, 223640)	128	718	1747.2	664	519	1747.2
Lyu(754309, 41192010)	128	2513	321812.6	1357	1153	321812.6
BG(145, 2204)	256	412	68.9	504	357	34.4
BG(561, 8512)	256	534	266	554	416	133
Lyu(6737, 102144)	256	822	3192	664	525	1596
Lyu(754309, 11435188)	256	3116	357349.6	1262	1186	178674.8

AES(C) 187cyc/block, AESNI 63cyc/block, Mersenne Twister 150cyc/128bit.

Observations

- Table size is reduced as expected when 256-bit precision
- CP Sampler with AESNI is fastest in all cases
- CP Sampler tends to be faster than IS when m is large
- Data type of tables may affect the speed: integer (CP) or floating-point number (IS).

Results

Speed is an average of 100,000 samples

Scheme(S,m)	prec.	Inversion		Count-then-Permute		
		speed	memory	speed C	speed NI	memory
BG(145, 1624)	128	437	25.4	480	351	25.4
BG(561, 6272)	128	478	245.6	553	406	245.6
Lyu(6737, 223640)	128	718	1747.2	664	519	1747.2
Lyu(754309, 41192010)	128	2513	321812.6	1357	1153	321812.6
BG(145, 2204)	256	412	68.9	504	357	34.4
BG(561, 8512)	256	534	266	554	416	133
Lyu(6737, 102144)	256	822	3192	664	525	1596
Lyu(754309, 11435188)	256	3116	357349.6	1262	1186	178674.8

AES(C) 187cyc/block, AESNI 63cyc/block, Mersenne Twister 150cyc/128bit.

Observations

- Table size is reduced as expected when 256-bit precision
- CP Sampler with AESNI is fastest in all cases
- CP Sampler tends to be faster than IS when m is large
- Data type of tables may affect the speed: integer (CP) or floating-point number (IS).

Results

Speed is an average of 100,000 samples

Scheme(S, m)	prec.	Inversion		Count-then-Permute		
		speed	memory	speed C	speed NI	memory
BG(145, 1624)	128	437	25.4	480	351	25.4
BG(561, 6272)	128	478	245.6	553	406	245.6
Lyu(6737, 223640)	128	718	1747.2	664	519	1747.2
Lyu(754309, 41192010)	128	2513	321812.6	1357	1153	321812.6
BG(145, 2204)	256	412	68.9	504	357	34.4
BG(561, 8512)	256	534	266	554	416	133
Lyu(6737, 102144)	256	822	3192	664	525	1596
Lyu(754309, 11435188)	256	3116	357349.6	1262	1186	178674.8

AES(C) 187cyc/block, AESNI 63cyc/block, Mersenne Twister 150cyc/128bit.

Observations

- Table size is reduced as expected when 256-bit precision
- CP Sampler with AESNI is fastest in all cases
- CP Sampler tends to be faster than IS when m is large
- Data type of tables may affect the speed: integer (CP) or floating-point number (IS).

Conclusions

Summary

- We present CP Sampler: a generic sampler for arbitrary discrete distribution
- It requires precomputation of expected $O(m \log N)$ time
- Its table size is precision-independent
- Hence table size could be reduced in high precision settings like cryptographic usages
- It can be faster than Inversion Sampler depending on parameters, because of its table size

Future work

- Full implementation including Precomputations
- Implementation with smaller parameters using 64-bit block ciphers
- Find applications other than lattice cryptography

References

- [BG14]: Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. In *CT-RSA*, volume 8366 of *Lecture Notes in Computer Science*, pages 28–47. Springer, 2014.
- [BKLPPRSV07]: Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.
- [BKP14]: Karl Bringmann, Fabian Kuhn, Konstantinos Panagiotou, Ueli Peter, and Henning Thomas. Internal DLA: Efficient Simulation of a Physical Growth Model - (Extended Abstract). In *ICALP (1)*, volume 8572 of *Lecture Notes in Computer Science*, pages 247–258. Springer, 2014.
- [Dev86]: Luc Devroye. *Non-Uniform Random Variate Generation*. Springer, 1986.
- [FT15]: Martin Farach-Colton and Meng-Tsung Tsai. Exact Sublinear Binomial Sampling. *Algorithmica*, 73(4):637–651, 2015.
- [Lyu12]: Vadim Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer, 2012.
- [Micc11]: Micciancio, “Lattice-Based Cryptography”, In *Encyclopedia of Cryptography and Security (2nd Ed.)*, pages 713–715. Springer, 2011.

Parameter Setting of Discrete Gaussian

Parameter choice (S and m)

- S: For each security level, we used σ and $S = \sqrt{2\pi}\sigma$ suggested in [Lyu12] and [BG14].
- m: m is determined by a security level n and the following lemma.

[Lyu12]Lemma 4.4 or [BG14]Lemma1

For any $\kappa > 0$,

$$Pr_{x \leftarrow \mathcal{D}_\sigma}(|x| > \kappa\sigma) \leq 2e^{-\frac{\kappa^2}{2}}.$$

where, $\mathcal{D}_\sigma$ is a discrete gaussian of center 0 and stadard deviation σ.

- E.g. when $\kappa = 13.5$ the probability is bounded by 2^{130}. Hence $\mathrm{m} = 2 \cdot 13.5 \cdot \sigma$ is reasonable when n=128.

Sampling from binominal distribution $B(N, p)$

Bringmann et.al. [BKP14]

- Exact sampler from $\mathcal{B}(N, 1/2)$ with $O(1)$ time

Farach-Colten and Tsai [FT15]

- Sampling from $\mathcal{B}(N, p)$ for arbitrary p
- $\mathcal{B}(N, 1/2)$ sampler is used as a black box
- Time complexity is
  - Expected $O(\log N)$ times of $\mathcal{B}(N, 1/2)$ call

  or
  - $O(1)$ time in high probability, with $O((\log N)^{\epsilon})$ time precomputation, for any positive ϵ
- Implemented around $N = 2^{30}$

\Orchestrating a brighter world

NEC