# Format-Preserving Encryption

Somitra Kumar Sanadhya

Indian Institute of Technology Ropar

August 27, 2020

# Credits

Credits for the work described:

- Co-authors (Designs): Donghoon Chang, Mohona Ghosh, Kishan Chand Gupta, Arpan Jati, Abhishek Kumar, Dukjae Moon, Indranil Ghosh Ray.
- Co-authors (Recent cryptanalysis results): Orr Dunkelman, Abhishek Kumar, Eran Lambooij.

- Significant contribution in preparing these slides: Abhishek Kumar, IIT Ropar.

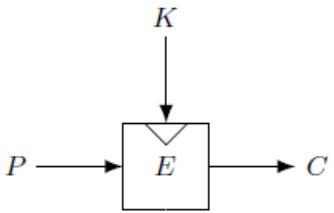
- Funding: India-Israel Collaborative Project, DST, Govt. of India.

The research described is a part of the (ongoing) PhD work of Abhishek.

# Block cipher

- Block cipher: A family of permutations indexed by the secret key.
- Deterministic primitive.
- Security notion: PRP.
- Modes of operation and padding schemes are required to construct an encryption scheme.
- Syntax:

$$E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$$

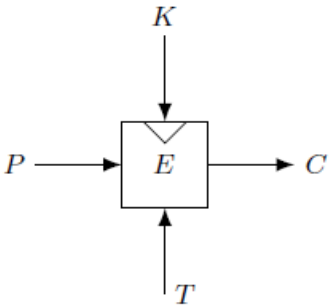



- Issues:
  - Can't be used for small domains (Codebook attack)
  - Changing the key is costly.

# Tweakable Block Cipher

- A block cipher with a twist - "tweak".
- A public tweak allows switching to a different family of permutations (even for the same key).
- Changing the tweak is a low cost operation.
- Syntax:

$$E : \{0,1\}^k \times \{0,1\}^n \times \{0,1\}^t \rightarrow \{0,1\}^n$$




- Idea used first in the "Hasty Pudding cipher" by Schroeppel (AES competition, 1998).
- Formalized by Liskov, Rivest, Wagner (Crypto 2002).

# Format Preserving Encryption (FPE)

- FPE: Encryption scheme where the input and the output have the same format.


- Why can't we use a block cipher?
  - Loss of format
  - Ciphertext expansion
  - "Cipher text ... bears roughly the same resemblance to plain text ... as a hamburger to a T-bone steak." (Brightwell and Smith, 1997).

# Is the problem interesting?

- Applications
  - Credit card encryption, SSN encryption, ...
  - Database Encryption
  - Data capturing devises used to capture data of specific format (PIN pads, used with ATM machines).
- Product
  - First product of Voltage Security Inc. (Now Microfocus Inc.)
- Standardization
  - Draft NIST SP 800-38G (March 2016), updated in 2017.
- Requirements
  - Any (user defined) format should be supported.
  - Ciphertext length expansion is not permitted.

First formal treatment by Bellare, Ristenpart, Rogaway, Stegers (Selected Areas in Cryptography 2009).

# Draft Standard

**Draft NIST Special Publication 800-38G**
**Revision 1**

## Recommendation for Block Cipher Modes of Operation

*Methods for Format-Preserving Encryption*

Morris Dworkin
*Computer Security Division*
*Information Technology Laboratory*

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-38Gr1-draft

February 2019

# FPE Security notions (BRRS'09)

- Pseudo-Random Permutation (PRP):
  - Distinguishing $E_k(\cdot)$ from $RP(\cdot)$.
- Single Point Indist. (SPI):
  - Distinguishing $E_k(m)$ (for adversarial choice of $m$) from a $c = E_k(r)$ for a random message $r$.
- Message Privacy (MP):
  - $E_k(m)$ reveals no information on $m$, except its format.
  - Formalized by comparing the "performance" of the real-world adversary to that of a degenerate adversary $S$ that can only make equality queries of the form "is $m$ the encrypted message?").
- Message Recovery (MR):
  - Adversary can't "completely reveal" $m$ when supplied with $E_k(m)$.

# Comments on the Security notions (BRRS'09)

- PRP $\implies$ all other security notions.
- SPI $\implies$ MP, MR with tight bound.
- MP and MR are what are needed in applications.

- PRP is an overkill.
- Attacks against PRP may not be a threat in practice for an FPE scheme.

# Existing FPE Schemes

- General techniques: (Black and Rogaway, CT-RSA 2002)
  - Prefix Cipher
    - Extension: Rank-then-Encipher (RtE) - BRRS'09
  - Cycle-walking
  - Generalized-Feistel Cipher
- Specific constructions:
  - FFSEM (Spies, Voltage Inc.) Superceded by FFX.
  - **FFX (FF1, FF2)**. (Bellare, Rogaway, Spies, Submitted to NIST, Feb 2010)
  - **BPS** (renamed FF3 by NIST). (Brier, Peyrin, Stern, Submitted to NIST, March 2010)
  - **VFPE** (John Sheets, Kim R. Wagner, VISA USA Inc., Submitted to NIST, Oct 2011)
  - **FEA-1 and FEA-2** (Lee, Koo, Roh, Kim, Kwon, ICISC 2014, Korean FPE Standard)

# General Technique 1: Prefix Cipher

- (Black and Rogaway, CT-RSA 2002)
- Domain $= \{0, 1, \ldots, t-1\}$.
- Use $n$-bit block cipher $E_k(.)$ with domain $N = 2^n \geq t$.
- Permut. $[0, 1, \ldots, t-1]$ = Ordering $[E_k(0), E_k(1), \ldots, E_k(t-1)]$.
- Method is computationally reasonable for small $t$ (such as $t < 2^{30}$).

# Extension: Rank-then-Encipher (RtE)

- "*It would be undesirable to design an encryption schemes whose internal workings were tailored to the specialized task in hand.*"
  – BRRS, SAC 2009.

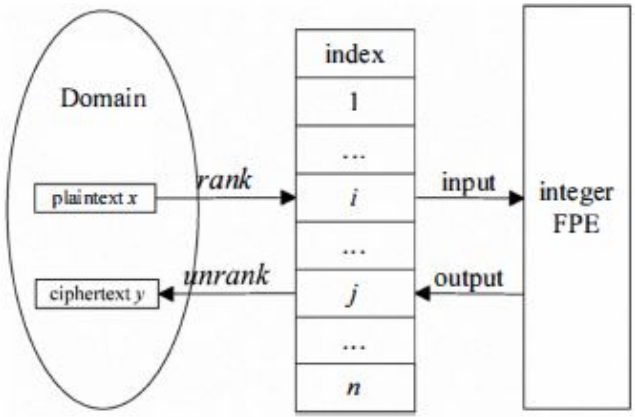
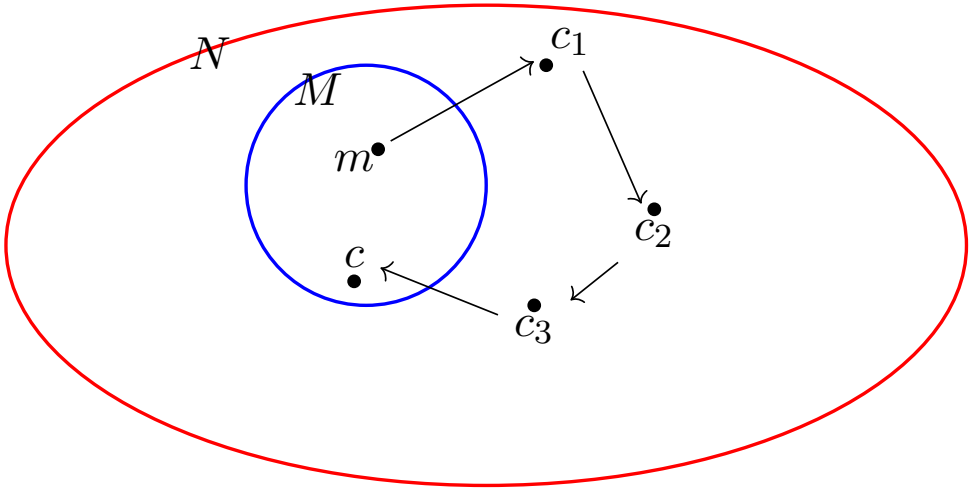



Figure 2: RtE scheme.

- integer domain FPE → arbitrary domain FPE.
- Given a format space, rank the input, and then use an integer FPE.
- If $E$ is secure then so is RtE($E$, rank, unrank).

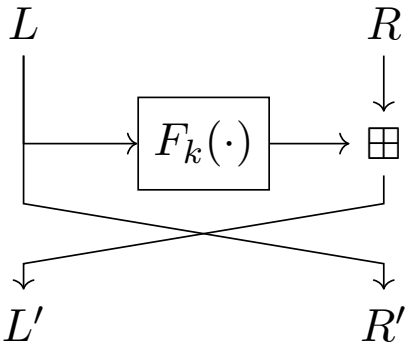
# General Technique 2: Cycle Walking

- (Black and Rogaway, CT-RSA 2002)
- Domain $M = \{0, 1, \ldots, m-1\}$.
- Use $E_k(.)$ with domain $N$ such that $|N| \geq |M|$.
- Map $m \in M$ to $E_k(\ldots E_k(\ldots E_k(m)) = c$ until $c \in M$.
- Need $M$ to be dense in block cipher domain $N$, otherwise too may block cipher invocations.



# General Technique 3: Generalized Feistel

- (Black and Rogaway, CT-RSA 2002)
- Let message space size $= t$.
- Choose two integers $a$ and $b$ such that $ab \geq t$, with $a \geq |L|$ and $b \geq |R|$ (Fig 4).




- Perform cycle walking when out of range.
- Efficient when $(ab - t)$ is small.
- Suggested number of rounds is 3.

Figure 4: One round of GF.

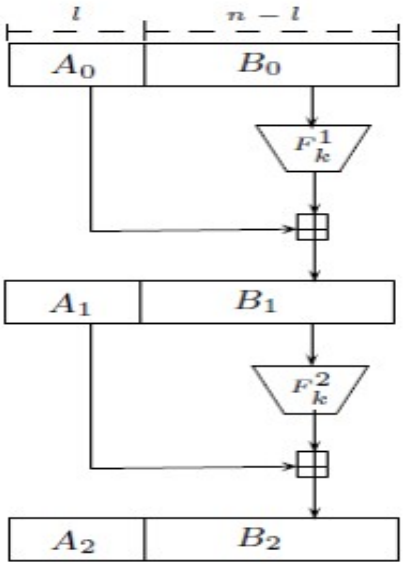
# Specific Construction 1: FFSEM

- Designed by Terrence Spies and submitted to NIST in 2008.
- Concrete instance of the Black-Rogaway technique.
- Uses a tweak to make round-PRFs different.
- Suggestions:
  - For message domain $> 40$ bits, use at least 6 rounds (considering Patarin's attack).
  - For message domain $\in \{32, 40\}$ bits, use extra rounds (not efficient for most of practical FPE applications).

# Specific Construction 2: FFX (FF1 and FF2)

- (BRRS 2009) "Theory" of FPE developed in this work.
- Two variants: (NIST Special Publication 800-38G)
  - Type-1 Feistel: FF1
  - Type-2 Feistel: FF2
- Both variants have at least 10 rounds of Feistel. (More, if message size or format is large).
- The round function is one invocation of AES.
- Thus, at least 10 calls to AES needed for each encryption or decryption.

Figure 5 and algorithm 1 represent two rounds and encryption function of FF1 respectively.




**Algorithm 1:** $\mathrm{FF1}_K^{N,T}(X)$

1 $(a, b) \leftarrow N; X_0 \leftarrow X$
2 for $i = 1, 2, \ldots, r(N)$ do
3 $\quad A_{i-1} \leftarrow X_{i-1} \text{ div } b$
4 $\quad B_{i-1} \leftarrow X_{i-1} \bmod b$
5 $\quad C_i \leftarrow (A_{i-1} + F_K(N, T, i, B_{i-1})) \bmod a$
6 $\quad X_i \leftarrow aB_{i-1} + C_i$
7 ret $X_{r(N)}$

Figure 5: Two Rounds of FF1.

# FF2

Figure 6 and algorithm 2 represent two rounds and encryption function of FF2 respectively.

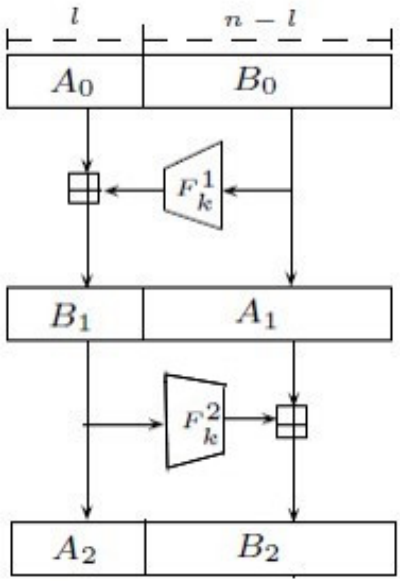



Figure 6: Two Rounds of FF2.

**Algorithm 2:** $\mathrm{FF2}_K^{N,T}(X)$

1 $(a, b) \leftarrow N$
2 $A_{i-1} \leftarrow X_{i-1}$ div $b$ ; $B_{i-1} \leftarrow X_{i-1}$ mod $b$
3 for $i = 1, 2, \ldots, r(N)$ do
4 if $i$ mod 2=1 then $s \leftarrow a$ else $s \leftarrow b$
5 $A_i \leftarrow B_{i-1}$
6 $B_i \leftarrow (A_{i-1} + F_K(N, T, i, B_{i-1}))$ mod $s$
7 ret $sA_{r(N)} + B_{r(N)}$

# Specific Construction 3: FF3

- Designed by Brier, Peyrin, Stern and named as BPS initially.
- It is a Feistel based design and consists of 8 rounds (faster than FFX).
- BPS is a combination of the following two components:
  1. A length restricted internal block cipher.
     - Initially renamed as FF3 by NIST, later renamed as FF3-1.
     - This internal block cipher is used to encrypt the data while preserving the format.
  2. A mode of operation to handle long messages.
     - This mode is malleable and hence not adopted by NIST.
     - Hence, maximum input size is fixed (unlike FF1, and FF2).
     - The tweak size is 64-bits, hence not suitable for very large messages.

# Analysis of FF1, FF2 and FF3

- (M. Dworkin and R. Perlner, eprint 2015/30): FF2 does not provide the expected 128-bits of security strength. Hence, removed from NIST recommended designs.
- (Bellare et. al, ACM CCS 2016): A practical attack message recovery attack for small domain messages for FF1 and FF3.
  – For one byte messages, the data complexity of the attacks is approx $2^{34}$ and $2^{42}$ for FF3 and FF1 respectively.
- (Hoang et. al, Crypto 2018) Improved attack to recover one byte message with data complexity $2^{27}$ and $2^{36}$ for FF3 and FF1 respectively.
- (Durak and Vaudenay, Crypto 2017) gave a generic attack against FF3 with complexity $O(N^{\frac{11}{6}})$ chosen plaintexts and time complexity $O(N^5)$, where $N^2$ is the domain size.
- All these attacks work only for messages of $\approx 15$ bits or smaller.

# Specific Constructions 4: FEA-1 & FEA-2

- Designed by a team of South Korean researchers in 2014.
- Currently a Korean FPE standard.
- Feistel based design, with a tweakable round function.
- The round function of FEA-1 and FEA-2 consists of two iteration of S-box layer and diffusion layer (like DES).

# Specific Constructions 4: FEA-1 & FEA-2

- Suggested number of rounds for FEA-1 and FEA-2 for different key sizes.

| Key length | FEA-1 | FEA-2 |
|---|---|---|
| 128 | 12 | 18 |
| 192 | 14 | 21 |
| 256 | 16 | 24 |

- As per designer's claim, both FEA-1 and FEA-2 are almost two times faster than FF1 and FF3.

# Status of FEA-1 and FEA-2

- No publicly known results.

# Summary

- Feistel based designs (FF1, FF2 and FF3) requires many invocation of internal primitives (due to Patarin's attack).
- Attack complexities for FF1 and FF3 (corresponding to the domain size $\{0, \ldots, 9\}^3$).

| Scheme | Design Strategy | Key Size | Attack Status | |
|---|---|---|---|---|
| | | | Exists | Complexity |
| FF1 | Feistel | 128 | Yes | $2^{58}$ |
| FF2 | Feistel | 128 | Yes | Dropped by NIST |
| FF3 | Feistel | 128 | Yes | $2^{21}$ |
| FEA-1/-2 | Feistel | 128/192/256 | No | – |

# New Designs

- Problems with existing designs:
  - All the existing designs are based on Feistel structure.
  - Due to Patarin's attacks, the number of Feistel rounds needed are high, thus efficient issues.
  - Large modular operations used in these designs, which makes them unsuitable for resource constrained devices.
- Can we have a different structure? And more efficient designs?

# Substitution-Permutation Network (SPN)

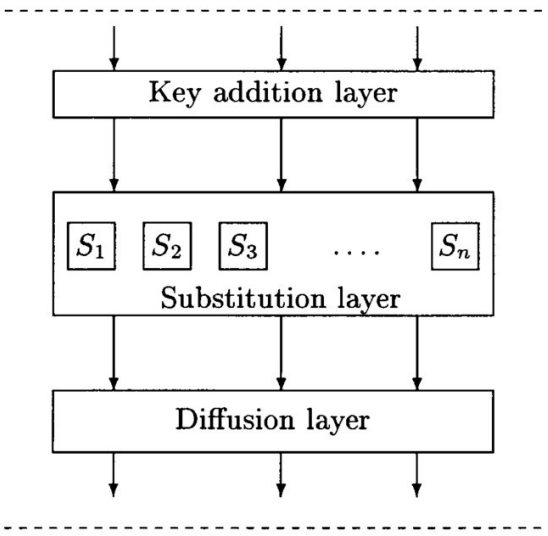



Figure 7: One round of SPN.

- Basic transformations: Key addition, confusion (substitution) and diffusion.
- A round transformations are combination of all these basic transformations.
- Finally the cipher consists iteration of many rounds.

# Designing FPE using SPN paradigm

FPE may be designed in several ways using SPN as follows:

1. Every transformation of SPN preserves the format.
   – Our design: SPF (Inscrypt 2016).
2. Each round preserves the format (but individual operation are not format preserving).
3. Entire cipher is format preserving (but each round does not necessarily preserves the format).
   – Our design: eSPF (HASS 2019).

We use SPN in counter mode to generate FP stream, which is then modularly added to the message to encrypt.

# The Operating Mode of SPF

- We adopt the Counter Mode of operation using SPF routine internally to handle variable lengths of messages.
- Generate $j$ blocks of plaintexts, i.e., $M = M_1||M_2||\ldots||M_j$ such that $\forall i, 1 \leq i < (j-1), |M_i| = n$ and $|M_j| \leq n$.
- Invoke SPF for each counter blocks $T_1, T_2, \ldots, T_j$ corresponding to the message blocks to generate corresponding ciphertext blocks $C_1, C_2, \ldots, C_j$.
- The chiphertext $C$ is concatenation of these blocks.

# Encryption process

- Block input = A counter made up of 16 symbols.
- Note that the entropy of the input is not necessarily 128 bits.
- Arrange the input in a $4 \times 4$ matrix.
- Apply operations like AES. Explained ahead.
- Output is a stream of symbols.
- Modular addition of the stream to input message produces the ciphertext.

# SPF

- '`SPF`' stands for Substitution-Permutation based, Format preserving encryption.
- Motivated by AES design.
- All basic transformations preserve the format, hence the cipher preserves the format.
- $\texttt{SPF}_r^N$, a member of `SPF` family consist $r$-rounds and works for the alphabet set $\Sigma$, where $|\Sigma| = N$.

# Basic Transformations

- Format-Preserving SubBytes (FPSB).
  – A bijective mapping form $\Sigma \rightarrow \Sigma$.
- ShiftRows.
  – Rotates the rows cyclically.
- Format-Preserving MixColumns (FPMC).
  – Explained at next slide.
- Format-Preserving Key Addition (FPKA).
  – Symbol wise modular addition of round-key and current state.
- Format-Preserving Tweak Addition (FPTA).
  –Symbol wise modular addition of round tweak and current state.

# FP MixColumn Transformation

- Binary Matrix: All elements are either 1 or 0.
- The binary matrix $\mathbf{M}$ used for presented scheme has branch number 4.

$$\mathbf{M} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

- The transformation of a column of state is represented as:

$$\begin{bmatrix} b_{r,0} \\ b_{r,1} \\ b_{r,2} \\ b_{r,3} \end{bmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \times \begin{bmatrix} a_{r,0} \\ a_{r,1} \\ a_{r,2} \\ a_{r,3} \end{bmatrix} \mod N$$

# SPF for Digits

- The motivating applications of `FPE` are `CCN` and `SSN`, i.e, string of digits.
- We present an instance of `SPF` construction for digits.

# S-boxes for Digits

- We use a single S-box $S : \Sigma \to \Sigma$, where $|\Sigma| = 10$.
- We analyzed all possible 10! mapping of $S$ exhaustively and picked up mappings that have good differential and linear probabilities.
- Finally, we used hardware implementation cost as the final metric to choose the S-box for digits
  - Using Karnaugh-map, we can compute the number of Boolean gates required for any mapping.

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| $S[x]$ | 2 | 6 | 8 | 7 | 1 | 4 | 9 | 5 | 3 | 0 |

Table 1: Representation of S-box for digits.

# Key Schedule

- We propose a new format preserving key scheduling algorithm (KSA) for the `SPF` cipher (Inscrypt 2016).
- The key schedule algorithm takes the 128-bit cipher key $K$ input and generates $(r + 1)$ round subkeys as outputs.

# Tweak Schedule

- Choice of tweak addition of `SPF` construction is motivated by the tweak addition of KIASU-BC cipher (Jean et. al, Asiacrypt 2014).
- An 16 symbol long tweak $Tw$ is used to generate two subtweaks $Tw0$ and $Tw1$. These are added to the first two rows of the state for each even and odd numbered rounds correspondingly.

# Security Analysis

We analyzed security of SPF against following attacks:

- Differential Attack.
- Linear Attack.
- Square Attack.
- Impossible Differential Attack.
- Key Related Attacks.
- Related Tweak Attacks.
- Multiset Attack.

Nine rounds of `SPF` are secure against these attacks. By considering performance and security margin, we recommend $r = 14$.

# Performance Analysis

- Efficiency comparison of FF1, FF3-1 and `SPF` for digits.

  Table 2: An entry corresponding to FPE schemes and number of symbols represent number of substitution and permutation required.

| FPE schemes | No. of Symbols | | | |
|---|---|---|---|---|
| | `16` | `32` | `64` | `128` |
| `SPF` | 14 | 28 | 56 | 112 |
| `FF1` | 110 | 110 | 310 | 410 |
| `FF3-1` | 80 | 80 | 160 | 240 |

- `SPF` is 5 to 7 times efficient than NIST standards for most of practical applications.

# Implementations

- We implemented $\texttt{SPF}_{14}^{10}$ on different 64-bit platforms and compared the performance with `FFX`.
- Similar to `AES`, table lookups in `SPF` can be used to combine different round transforms, leading to very fast implementations.
- Interestingly, the lookup tables are much smaller compared to `AES`.
- Apart from making such implementations more interesting for resource constrained environments, the small lookup tables also reduce risk of timing attacks.
- Unlike `AES`, `SPF` uses a binary matrix for the FPMC operation, as a result the implementation can be much more efficient on constrained devices.

# Implementations

We tested our implementation on high performance platforms, for server as well as mobile platforms, the experimental results are shown below.

Table 3: Experimental results on various 64-bit processors for digits.

| Processor | Clock Speed | Speed for $\texttt{SPF}_{14}^{10}$ | |
|---|---|---|---|
| | | Symbols/s. | Cycles/Symbol |
| Intel Core i7 6700 | 3.4 GHz | $132.4 \times 10^6$ | 25.6 |
| Intel Core i7 4770 | 3.4 GHz | $117.2 \times 10^6$ | 29.0 |
| Intel Core i5 5200 | 2.2 GHz | $26.8 \times 10^6$ | 82.0 |
| Intel Xeon E5 2630 | 2.3 GHz | $52.9 \times 10^6$ | 43.4 |

# Limitations

- Uses binary matrix in diffusion layer to meet the goal of designing format preserving basic transformations.
  – Non optimal choice due to non-availability of MDS matrix.
- Need to design specific basic transformations for each format-sizes.
  – One instantiation will not work for other formats.
- This construction will not work for some specific formats.

# Availability of MDS matrix for some specific formats

## Lemma 1

*Let $M = (m_{i,j})$ be a $d \times d$ MDS matrix over a field $\mathbb{F}_{2^b}$. Let $S$ be a set of $v$ elements such that $S = \{0, 1, \ldots, v-1\}$; and $S^* = S - \{0\}$. Further, let $S$ be a subset of $\mathbb{F}_{2^b}$ such that $\{0, 1\} \subset S$ and $\forall X \in S^d$, $Y = M \times X^T \in S^d$. Then :*

1. $m_{i,j} \in S^*$, $\forall\ 0 \leq i, j \leq d$,
2. $S^* = m_{i,j} S^*$, $\forall\ 0 \leq i, j \leq d$,
3. *For any $i, j \in \{0, \ldots, d-1\}$, the cyclic group generated by $m_{i,j}$ with respect to multiplication is subset of $S^*$, i.e., $\langle m_{i,j} \rangle \subseteq S^*$,*
4. *$S$ is an additive subgroup of $\mathbb{F}_{2^b}$.*

- The above lemma shows existence of a $d \times d$ format-preserving MDS matrix exists if $|S| = v$ is of the form $2^b$.

# Non-availability of MDS matrix for an arbitrary format

- (Gupta et al., Indocrypt 2016) It is impossible to construct any cryptographically significant $4 \times 4$ matrix over the field $\mathbb{F}_{2^4}$ in the diffusion layer which yields a format preserving set of cardinality 10.
- (Barua et al., Indocrypt 2018) Method to construct format preserving sets of cardinality 20 with respect to $3 \times 3$ MDS matrices; and of cardinality $10^3$ and $26^3$ with respect to $4 \times 4$ MDS matrices.
- They leave finding any format preserving sets of cardinality 10 and 26 with respect to $4 \times 4$ MDS matrices as open problems.

# eSPF

- We present a new approach to construct efficient format-preserving encryption schemes.
- Use of MDS matrix in the diffusion layer (unlike `SPF` construction where binary matrix is used) results in optimal diffusion.
- The proposed construction works for any domain size.
- One instantiation may work for many formats.

# eSPF Construction

The `eSPF` cipher contains two components:

- A non format preserving encryption.
  – The MDS matrix and substitution mappings are defined over the finite field $\mathbb{F}_{p^b}$, where $p$ is a prime and $b$ is an integer and $N \leq p^b$.

- A *Discarding Algorithm (DA)*.
  – In order to retain the format preserving property.

# Basic Transformations

- SubBytes (SB).
  – A bijective mapping $S : \mathbb{F}_{p^b} \rightarrow \mathbb{F}_{p^b}$.
- ShiftRows (SR).
  – Rotates symbols of the rows cyclically.
- MixColumns (MC).
  – A MDS matrix defined over $\mathbb{F}_{p^b}$ such that $N \leq p^b$.
- Key Addition (KA).
  – Symbol wise modular addition of round-key and the current state.
- Tweak Addition (TA).
  – Symbol wise modular addition of round tweak and the current state.

# Discarding Algorithm

- As the operations are performed over $\mathbb{F}_{p^b}$, we need to discard symbols which are not in format.
- The discarding process is equivalent of cycle-walking or using modular operation to ensure non-violation of the format.

**Algorithm 3:** DA($S$)

**input :** String $S$
**output:** String $S'$

Initialize a string $S'$=NULL;
For $i \leftarrow 1$ **to** $n$
  **if** $S[i] \in \Sigma$
    $S' = S'||S[i]$;
  **else**
    $S'$ ;
**return** $S'$;

# Discarding rate

- Let $\Sigma = \{0, 1, 2, \ldots, N-1\}$ be the alphabet set of format size $N$.
- Let $\Sigma' = \{0, 1, 2, \ldots, N'-1\}$, where $\Sigma'$ is the alphabet set containing all the elements of $\mathbb{F}_{p^b}$ and $N' > N$.
- Let $p_r$ be the probability of not discarding an output symbol of $E_k$, i.e., $p_r = \frac{N}{N'}$.
- As the ciphertext state consists of 16 symbols, the likelihood of getting a format compliant symbol 16 times can be modelled using binomial distribution.

# Discarding rate

In the following table, we are mentioning the candidate Galois Field for formats of different sizes with discarding details.

Table 4: Galois Field Size for formats of different sizes.

| Use Case | Format Size (N) | $\mathbb{F}_{p^b}$ | Field Size (N') | $p_r$ | Exp. Val.($\mu$) (for $x = 16$) | Binom. Pr. Pr.[$Z \geq 13$] |
|---|---|---|---|---|---|---|
| Digits | 10 | $\mathbb{F}_{11}$ | 11 | 0.9 | 14.54 | 0.94 |
| Alphabets | 26 | $\mathbb{F}_{29}$ | 29 | 0.89 | 14.24 | 0.9 |
| Case sensitive alphanumeric | 36 | $\mathbb{F}_{37}$ | 37 | 0.97 | 15.5 | 0.98 |
| Case insensitive-alphanumeric | 62 | $\mathbb{F}_{67}$ | 67 | 0.92 | 14.8 | 0.96 |

# S-boxes for Digits

- We use a single S-box $S$: $\mathbb{F}_{11} \rightarrow \mathbb{F}_{11}$.
- We analyzed all possible 11! mapping of $S$ exhaustively and picked up mappings that have good differential and linear probabilities.
- Finally, we used hardware implementation cost as the final metric to choose the S-box for digits
  - Using Karnaugh-map, we can compute the number of Boolean gates required for any mapping.

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $S[x]$ | 2 | 0 | 10 | 6 | 3 | 8 | 9 | 4 | 7 | 5 | 1 |

Table 5: Representation of S-box.

# MixColumns Transformation

- We choose a 4 x 4 MDS diffusion layer over GF(11) such that it violates the format size.
- The matrix $M$ used for the presented scheme has a branch number 5.

$$M = \begin{pmatrix} 1 & 1 & 2 & 5 \\ 5 & 1 & 1 & 2 \\ 2 & 5 & 1 & 1 \\ 1 & 2 & 5 & 1 \end{pmatrix}$$

# Security Analysis

We analyzed security of eSPF against the following attacks:

- Differential Attack.
- Linear Attack.
- Square Attack.
- Impossible Differential Attack.
- Key Related Attacks.
- Related Tweak Attacks.

Seven rounds of `eSPF` are secure against these attacks. By considering performance and security margin, we recommend $r = 10$.

# Implementations

- `eSPF` is suitable for efficient implementation on a wide range of devices.
- We implemented $\texttt{eSPF}_{10}^{10}$ on 64-bit platforms and compared the performance with `FFX`.
- Similar to `AES`, table lookups in `eSPF` can be used to combine different round transforms, leading to very fast implementations.
- Interestingly, the lookup tables are much smaller compared to `AES`.
- Apart from making such implementations more interesting for resource constrained environments, the small lookup tables also reduce risk of timing attacks.

# Performance Analysis

- Efficiency comparison of FF1, FF3-1, `SPF` and `eSPF` for digits.

Table 6: An entry corresponding to FPE schemes and number of symbols represent number of substitution and permutation required.

| FPE schemes | No. of Symbols | | | |
|---|---|---|---|---|
| | `8` | `16` | `32` | `64` |
| `eSPF` | 10 | 20 | 30 | 50 |
| `SPF` | 14 | 14 | 28 | 56 |
| `FF1` | 110 | 110 | 110 | 310 |
| `FF3-1` | 80 | 80 | 80 | 160 |

# Implementations

- We tested our implementation on high performance platforms, for server as well as mobile platforms, the experimental results are shown in Table 7.

Table 7: Experimental results on various 64-bit processors for digits.

| Processor | Clock Speed | Speed for $\texttt{eSPF}^{10}_{10}$ | |
|---|---|---|---|
| | | Symbols/s. | Cycles/Symbol |
| Intel Core i7 6700 | 3.4 GHz | 201.2 $\times 10^6$ | 16.8 |
| Intel Core i7 4770 | 3.4 GHz | 168.1 $\times 10^6$ | 20.2 |
| Intel Core i5 2400 | 3.1 GHz | 44.8 $\times 10^6$ | 70.5 |

- For other format sizes, performance would be very similar, till the *lookup-tables* can be kept in the L1 cache.
- Hardware implementation of `eSPF` is almost 10 times faster than the software implementations considering the clock frequency.

# Attacks on FF1, FF3-1, FEA-1 and FEA-2

- Joint work with Eran Lambooij, Orr Dunkelman, Abhishek Kumar.

- Supported by MOST and DST.

# Generic Construction

- The design of a block cipher over small domains has two inherent problems:
  1. Enumeration attacks.
  2. Constructing good components that work for a wide range of input sizes is not trivial.
- One round of generic construction of FPE designs.

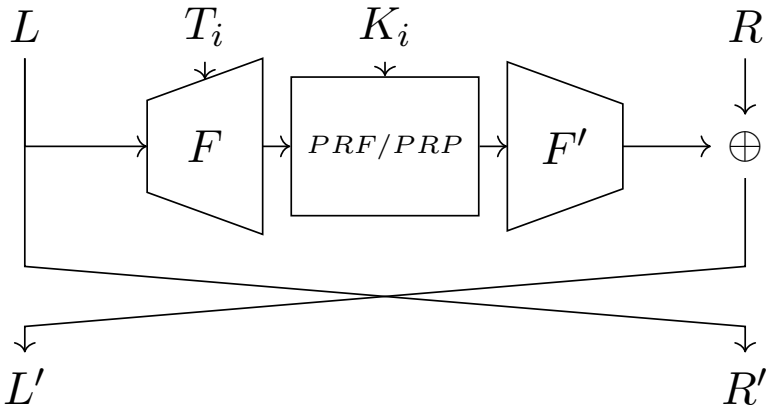



Figure 8: One round of the construction

# Security Analysis of FF1 and FF3

- We show how to construct a generic differential distinguisher for FF1 and FF3-1.
- This distinguisher is independent of the components used in the constructions and can be used for every cipher using an equivalent construction.
- We show that the amount of data needed to mount the attack is well within the security bounds of both FF1 and FF3-1.

# Distinguishing Attack

For a PRF:

- Given a non-zero difference $\Delta$, we can get a zero output difference.
- Using this property we can construct the following two round iterative differential characteristic with probability $2^{-n}$.

$$(0|\Delta) \xrightarrow{1} (\Delta|0) \xrightarrow{2^{-n}} (0|\Delta)$$

# Distinguishing Attack

- We argue that if the characteristic given above is not followed, the probability that we see the output difference $(\Delta|0)$ is $2^{-2n}$.
- Thus the probability of seeing this output when using an FFX like scheme with $2r$ rounds is $2^{-2n} + 2^{-rn}$.
- In other words, there is a $2^{-rn}$ positive bias to observe the output difference $(0|\Delta)$.

# Distinguishing Attack

- Normally, a bias 0f $2^{-rn}$ would not be distinguishable in a cipher with a blocksize of size $2^{2n}$ (for $r > 1$).
- However, the addition of the tweak in the construction increases the amount of samples we can gather using a single key.

# Attack Idea for FEA-1 and FEA-2

- The basic idea is same as discussed for FF1 and FF3-1.
- We may construct the following two round iterative differential characteristic with probability $2^{-n}$.

$$(0|\Delta) \xrightarrow{1} (\Delta|0) \xrightarrow{2^{-n}} (0|\Delta)$$

# Complexity of the Attack

- The number of rounds specified for each key size and the data needed to distinguish the permutation from random.

| | | | | Complexity | |
|---|---|---|---|---|---|
| Algorithm | Rounds | Block size | Keysize | Time | Data |
| FEA1 | 12 | 8 | 128 | $2^{36}$ | $2^{32}$ |
| FEA1 | 14 | 8 | 192 | $2^{44}$ | $2^{40}$ |
| FEA1 | 16 | 8 | 256 | $2^{52}$ | $2^{48}$ |
| FEA2 | 18 | 8 | 128 | $2^{60}$ | $2^{56}$ |
| FEA2 | 21 | 8 | 192 | $2^{72}$ | $2^{68}$ |
| FEA2 | 24 | 8 | 256 | $2^{84}$ | $2^{80}$ |
| FF1 | 10 | 20 | 128 | $2^{70}$ | $2^{60}$ |
| FF3-1 | 8 | 40 | 128 | $2^{100}$ | $2^{80}$ |
| Generic | $2r$ | $2n$ | - | $2^{2n(r-1.5)}$ | $2^{2n(r-1.5)-n}$ |

Table 8: Comparison of distinguishing attacks.

# Key Recovery Attack

- FF1 and FF3-1 use AES as the round function, and hence key recovery attack is not possible.
- However, the round function of the FEA-1 and FEA-2 is cryptographically weaker than AES.
- Given the above mentioned distinguisher, key recovery attacks against all key-sizes are possible for FEA-1 and FEA-2 (details skipped).

# References I


1. Morris Dworkin. Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption. NIST Special Publication, 800:38G.
2. J. Daemen, V. Rijmen, AES proposal: Rijndael, in NIST AES Proposal (1998).
3. J. Black and P. Rogaway. Ciphers with arbitrary finite domains. In B. Preneel, editor, CT-RSA 2002.
4. Schroeppel, R.: The Hasty Pudding Cipher (1998).
5. Liskov et. al, Tweakable Block Ciphers. In: Yung, M. (ed.) CRYPTO 2002.
6. Gupta et. al, Format Preserving Sets: On Diffusion Layers of Format Preserving Encryption Schemes. INDOCRYPT 2016.

# References II


7. Barua et. al, On Diffusion Layers of SPN Based Format Preserving Encryption Schemes: Format Preserving Sets Revisited. INDOCRYPT 2018.
8. Lee et. al, Format-preserving encryption algorithms using families of Tweakable Blockciphers. In: Lee J, Kim J (eds). ICISC 2014.
9. Chang et. al, A Generalized Format Preserving Encryption Framework Using MDS Matrices. Journal of Hardware and Systems Security, Vol. 3, 2019, Springer.
10. Chang et. al, SPF: A New Family of Efficient Format-Preserving Encryption Algorithms, Inscrypt 2016.
11. M. Dworkin and R. Perlner, Analysis of VAES3 (FF2), Report no. 2015/306, IACR Cryptology ePrint Archive, April 2, 2015.

12 Bellare at. al, Message-recovery attacks on feistel-based format preserving encryption, ACM CCS 2016.

# References III


13 F. B. Durak and S. Vaudenay, Breaking the FF3 Format-Preserving Encryption Standard Over Small Domains, CRYPTO 2017.

14 Hoang et. al, The Curse of Small Domains: New Attacks on Format-Preserving Encryption, CRYPTO 2018.

15 Jean et.al, Tweaks and keys for block ciphers: the TWEAKEY framework, ASIACRYPT 2014.

# THANK YOU