E2 - A Candidate Cipher for AES

Masayuki Kanda, Shiho Moriai, Kazumaro Aoki, Hiroki Ueda, Miyako Ohkubo, Youichi Takashima, Kazuo Ohta, Tsutomu Matsumoto*

e2@isl.ntt.co.jp
http://info.isl.ntt.co.jp/e2/

Nippon Telegraph and Telephone Corporation (NTT)
*Yokohama National University
Outline

• Overview
• Design
• Security
• Performance
• Conclusion
Design Goals

- A 128-bit symmetric block cipher
- Key length of 128, 192, and 256 bits
- Security: secure against all known attacks and more
- Efficiency: faster than DES
- Flexibility: efficient implementations on various platforms
Security of E2 (1)

There are many attacks….
Security of E2 (1)

Brute Force Attacks

There are many attacks....
There are many attacks....
Security of E2 (1)

There are many attacks....

Brute Force Attacks
Differential Cryptanalysis
Linear Cryptanalysis
Security of E2 (1)

There are many attacks....

- Brute Force Attacks
- Differential Cryptanalysis
- Linear Cryptanalysis
- Higher Order Differential Attack
Security of E2 (1)

- Brute Force Attacks
- Differential Cryptanalysis
- Linear Cryptanalysis
- Higher Order Differential Attack
- Interpolation Attack

There are many attacks....
There are many attacks....

Security of E2 (1)

- Brute Force Attacks
- Differential Cryptanalysis
- Linear Cryptanalysis
- Higher Order Differential Attack
- Interpolation Attack
- Partitioning Cryptanalysis
Security of E2 (2)

E2 is proven to have sufficient security
Security of E2 (3)

- Differential Cryptanalysis
- Linear Cryptanalysis
- Higher Order Differential Attack
- Interpolation Attack
- Partitioning Cryptanalysis

S-box is designed to have no vulnerabilities
Security of E2 (4)

E2 supports 128-bit block size and 128,192, 256-bit key sizes.
Design Goals (cont.)

- A 128-bit symmetric block cipher
- Key length of 128, 192, and 256 bits
- Security: secure against all known attacks and more
- Efficiency: faster than DES
- Flexibility: efficient implementations on various platforms
### Efficiency and Flexibility of E2

<table>
<thead>
<tr>
<th>Processor</th>
<th>Frequency</th>
<th>Clocks/Block</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>200MHz Intel Pentium Pro</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANSI C (Borland C++ 5.02)</td>
<td>711</td>
<td>36.0</td>
<td></td>
</tr>
<tr>
<td>Assembly</td>
<td>420</td>
<td>61.0</td>
<td></td>
</tr>
<tr>
<td><strong>600MHz DEC 21164A</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assembly</td>
<td>600</td>
<td>128</td>
<td></td>
</tr>
<tr>
<td><strong>5MHz Hitachi H8/300</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assembly</td>
<td>6,374</td>
<td>100.5</td>
<td></td>
</tr>
</tbody>
</table>
Efficiency and Flexibility of E2

200MHz Intel Pentium Pro
ANSI C
(Borland C++ 5.02)
711 clocks/block
36.0 Mbits/sec

32-bit CPU
64-bit CPU

600MHz DEC 21164A
Assembly
600 clocks/block
128 Mbits/sec
5MHz Hitachi H8/300
Assembly
6,374 clocks/block
100.5 k bits/sec

cf. DES (RSAREF, Borland C++ 5.0)
10.6 Mbits/sec
Outline

• Overview
• Design
• Security
• Performance
• Conclusion
High-level Structure of E2

Plaintext P → Key Scheduling Part

Key K → Ciphertext C
High-level Structure of E2

Plaintext $P$  Key $K$

Data randomizing part

Ciphertext $C$
High-level Structure of E2

Plaintext $P$  Key $K$

Data randomizing part

Key scheduling part

Ciphertext $C$
Data Randomizing Part Framework

- **IT-Function**
  (Initial Transformation)

- Feistel structure

- **FT-Function**
  (Final Transformation)
Design Rationale of Framework

• Feistel structure
  ◆ Widely known and thought to offer long-term security
  ◆ Symmetric encryption and decryption
  ◆ Evaluation of security against DC and LC has been well studied

• IT-Function and FT-Function
  ◆ Offer a proactive design and hinder later attacks
Design Rationale of Framework

- Feistel structure
  - Widely known and thought to offer long-term security
  - Symmetric encryption and decryption
  - Evaluation of security against DC and LC has been well studied

- \textit{IT}-Function and \textit{FT}-Function
  - Offer a proactive design and hinder later attacks
Design Rationale of F-Function (1)

- Structures for which security evaluation against DC and LC is easy
  - 1-round SPN structure (e.g., DES)
  - Recursive structure (e.g., MISTY)
  - 2-round SPN structure
- Comparing the speed at the same level of security, we decided to adopt 2-round SPN structure
Design Rationale of F-Function (1)

- Structures for which security evaluation against DC and LC is easy
  - 1-round SPN structure (e.g., DES)
  - Recursive structure (e.g., MISTY)
  - 2-round SPN structure

- Comparing the speed at the same level of security, we decided to adopt 2-round SPN structure
**Design Rationale of F-Function (1)**

- Structures for which security evaluation against DC and LC is easy
  - 1-round SPN structure (e.g., DES)
  - Recursive structure (e.g., MISTY)
  - 2-round SPN structure

- Comparing the speed at the same level of security, we decided to adopt 2-round SPN structure

Evaluated using practical measure
Practical Measure for Feistel Cipher

• General case [Knudsen (FSE’93)]
  ✷ Number of rounds: \( R = 2r, 2r + 1 \)
  ✷ Evaluation: \( UDCP^{(R)} = p^r, \quad ULCP^{(R)} = q^r \)

• Bijective case [Kanda et al. (SAC’98)]
  ✷ Number of rounds: \( R = 3r, 3r + 1, 3r + 2 \)
  ✷ Evaluation: \( UDCP^{(R)} = p^{2r}, \quad ULCP^{(R)} = q^{2r} \)
  \( (R = 3r, 3r + 1) \)
  \( UDCP^{(R)} = p^{2r+1}, \quad ULCP^{(R)} = q^{2r+1} \)
  \( (R = 3r + 2) \)

Note: \( p, q \): Maximum differential and linear prob. of round function
Practical Measure for Feistel Cipher

- General case [Knudsen (FSE'93)]
  - Number of rounds: \( R = 2r, 2r + 1 \)
  - Evaluation: \( \text{UDCP}(R) = p^r, \text{ULCP}(R) = q^r \)

- Bijective case [Kanda et al. (SAC'98)]
  - Number of rounds: \( R = 3r, 3r + 1, 3r + 2 \)
  - Evaluation:
    \[
    \text{UDCP}(R) = p^{2r}, \quad \text{ULCP}(R) = q^{2r} \quad (R = 3r, 3r + 1)
    \]
    \[
    \text{UDCP}(R) = p^{2r+1}, \quad \text{ULCP}(R) = q^{2r+1} \quad (R = 3r + 2)
    \]

Note: \( p, q \): Maximum differential and linear prob. of round function

When \( R = 6 \)
- \( \text{UDCP} = p^3 \) [General]
- \( \text{UDCP} = p^4 \) [Bijective]
Design Rationale of F-Function (2)

Performance vs. Secure

- Slow
- Fast

Graph showing:
- Total number of s-boxes (y-axis)
- Upper bounds of max differential/linear prob (log2) (x-axis)
- Security levels

- 1-round SPN structure
- Recursive structure
- 2-round SPN structure

Key points:
- 42 rounds: Secure
- 6 rounds: Not secure
- 9 rounds: Not secure

First AES Candidate Conference
Copyright NTT 1998
$F$ - Function Overview

$K^{(2)}$

$F$ - Function

$K^{(1)}$

$S$ - Function

$P$ - Function

$S$ - Function
Design Rationale of P-Function

- Maximize minimum number of active s-boxes
  - Minimize upper bound of maximum differential / linear prob. of round function
- Use only XOR operation
  - Simple construction
  - Efficient implementations in both software and hardware
- Minimize gate counts required for hardware
Design Rationale of P-Function

- Maximize minimum number of active s-boxes
  - Minimize upper bound of maximum differential / linear prob. of round function
- Use only XOR operation
  - Simple construction
  - Efficient implementations in both software and hardware
- Minimize gate counts required for hardware
# of Active s-boxes = 3 (Bad P-Function)

Many active s-boxes mean high security against DC.
# of Active s-boxes $\geq 5$ (E2 P-Function)
# of Active s-boxes ≥ 5 (cont.)
Design Rationale of s-box

1. Suitability for various platforms

2. No trap-doors

3. No vulnerability to known attacks
**Rationale 1 : Suitability for Various Platforms**

- **Table-lookup**
  - efficiency does not depend on processors with various word-lengths (8, 16, 32, 64 bits)

- **One 8-by-8-bit s-box**
  - consideration for 8-bit smart card implementations
Rationale 2: No trap-doors

- Design principle is publicly given
- Based on well-known mathematical functions
Candidates of s-box

- $s : \text{GF}(2)^8 \rightarrow \text{GF}(2)^8 ; x \mapsto s(x) = g(f(x))$

candidates of $f(x)$ and $g(x)$

I. $x^k$ in $\text{GF}(2^8)$ $\forall k \in \text{GF}(2^8), k \neq 1$

II. $u^x$ in $\mathbb{Z}/(2^8+1)\mathbb{Z}$ $\forall u \in \mathbb{Z}/(2^8+1)\mathbb{Z}, u \neq 0,1$

III. $x^k$ in $\mathbb{Z}/(2^8+1)\mathbb{Z}$ $\forall k \in \mathbb{Z}/(2^8+1)\mathbb{Z}, k \neq 1$

IV. $ax+b$ in $\mathbb{Z}/(2^8)\mathbb{Z}$ $\forall a, b \in \mathbb{Z}/(2^8)\mathbb{Z}$

V. $ax+b$ in $\mathbb{Z}/(2^8+1)\mathbb{Z}$ $\forall a, b \in \mathbb{Z}/(2^8+1)\mathbb{Z}$

$$3 \leq w_H(a), w_H(b) \leq 5$$

Note that $256 \in \mathbb{Z}/(2^8+1)\mathbb{Z}$ corresponds to $0 \in \text{GF}(2)^8$. 
Rationale 3: No Vulnerability to Known Attacks

- Considered Attacks
  - Differential cryptanalysis [BS90]
  - Linear cryptanalysis [M93]
  - Higher order differential attack [JK97]
  - Interpolation attack [JK97]
  - Partitioning cryptanalysis [HM97]
How to select s-box

• \( s : \text{GF}(2)^8 \rightarrow \text{GF}(2)^8 \); \( x \mapsto s(x) = g(f(x)) \)

I. \( f(x) = x^e \) in \( \text{GF}(2^8) \)

IV. \( g(y) = ay + b \) in \( \mathbb{Z}/(2^8)\mathbb{Z} \)

Composition of functions from different groups

expected to be effective in thwarting algebraic attacks, e.g., interpolation attack
How to select s-box parameters (1)

\[ s : \text{GF}(2)^8 \rightarrow \text{GF}(2)^8 ; \ x \mapsto s(x) = g \left( f(x) \right) \]

\[ f(x) = x^e \text{ in } \text{GF}(2^8) \]
\[ g(y) = ay + b \text{ in } \mathbb{Z}/(2^8)\mathbb{Z} \]

- Criteria for the considered 5 attacks
- Bijectivity
- Hamming weight of \(a, b\)
- Differential-linear prob.
How to select s-box parameters (2)
How to select s-box parameters (2)
How to select s-box parameters (2)

\[ 0 \leq e < 256 \]
\[ 0 \leq a, b < 256 \]
\[ 3 \leq w_H(a) \leq 5 \]
\[ 3 \leq w_H(b) \leq 5 \]
How to select s-box parameters (2)

$0 \leq a, b < 256$

3 $\leq w_1(a) < 5$

3 $\leq w_1(b) < 5$

0 $\leq e < 256$

$w_1(e) = 7$

$(a, b) = \{(97, 97), (97, 225), (225, 97), (225, 225)\}$

bijective?

$\alpha, \beta, e$

$32$ candidates survived
How to select s-box parameters (2)

\[ e : (255, e) = 1 \]

\[ a : \text{odd} \]
How to select s-box parameters (2)

\[ w_H(e) = 7 \]

\( w_H(e) = 7 \)

\( (a, b) = \{(97,97),(97,225), (225,97),(225,225)\} \)

32 candidates survived
How to select s-box parameters (2)

$q_s : \text{min } ?$

Linear Probability
How to select s-box parameters (2)

\( p_s : \text{min?} \)

Differential Probability
How to select s-box parameters (2)

$\mathbf{r_s : min ?}$

Differential-linear Prob.
How to select s-box parameters (2)

\[ w_H (e) = 7 \]

\[(a, b) = \{(97,97), (97,225), (225,97), (225,225)\} \]
How to select s-box parameters (2)

Higher Order Differential Attack

deg s : max?

3 <= \( w_2(f) \leq 5 \)
3 <= \( w_1(f) \leq 5 \)
0 <= \( w \leq 256 \)

\( w_2(e) = 7 \)
\( (a, b) = \{ (97, 97), (97, 225), (225, 97), (225, 225) \} \)

\( w_1(e) = 7 \)
\( (a, b) = \{ (97, 97), (97, 225), (225, 97), (225, 225) \} \)

32 candidates survived
How to select s-box parameters (2)

I_s ((F, G)) : small ?

Partitioning Cryptanalysis
How to select s-box parameters (2)

**Interpolation Attack**

**coeff \(_{2^8}^8 s : large?**

- \( 0 \leq a, b < 256 \)
- \( 3 \leq w_F(a) \leq 5 \)
- \( 3 \leq w_F(b) \leq 5 \)
- \( 0 \leq c < 256 \)

\( w_F(c) = 7 \)

\( (a, b) = \{(97,97),(97,225),(225,97),(225,225)\} \)

**Decision Points**

- \( a < min? \)
  - \( Y \) : discard \( a, b, c \)
- \( a > min? \)
  - \( N \) : discard \( a, b, c \)

- \( b < min? \)
  - \( Y \) : discard \( a, b, c \)
- \( b > min? \)
  - \( N \) : discard \( a, b, c \)

- \( c < min? \)
  - \( Y \) : discard \( a, b, c \)
- \( c > min? \)
  - \( N \) : GOAL

32 candidates survived
How to select s-box parameters (2)

\[
\text{coeff}_p s : \text{large} \quad ?
\]

\[
p: \text{prime s.t.} \quad 2^8 < p < 2^9
\]
How to select s-box parameters (2)
How to select s-box parameters (3)

$s : \text{GF}(2)^8 \rightarrow \text{GF}(2)^8 ; x \mapsto s(x) = g \left( f \left( x \right) \right)$

$f \left( x \right) = x^e \quad \text{in} \quad \text{GF}(2^8)$

$g \left( y \right) = ay + b \quad \text{in} \quad \mathbb{Z}/(2^8)\mathbb{Z}$

$e = 127, 191, 223, 239, 247, 251, 253, 254$

$(a, b) = (97, 97), (97, 225), (225, 97), (225, 225)$
How to select s-box parameters (3)

\[ s : \text{GF}(2)^8 \rightarrow \text{GF}(2)^8 ; x \mapsto s(x) = g (f(x)) \]

\[ f(x) = x^e \quad \text{in} \quad \text{GF}(2^8) \]

\[ g(y) = ay + b \quad \text{in} \quad \mathbb{Z}/(2^8)\mathbb{Z} \]

\[ e = 127, 191, 223, 239, 247, 251, 253, 254 \]

\[ (a, b) = (97, 97), (97, 225), (225, 97), (225, 225) \]

\[ (a, b, e) = (97, 225, 127) \text{ was selected.} \]
**High-level Structure of E2**

Plaintext $P$  
Key $K$

- **Data randomizing part**
- **Key scheduling part**

Ciphertext $C$
Design Rationale of IT / FT-Functions

Goal: To protect $E_2$ against future advances in cryptanalysis

$IT$-Function: avoid linking plaintext to inputs to first $F$-Function

$FT$-Function: avoid linking ciphertext to outputs from last $F$-Function
IT-Function and FT-Function Overview

IT Function:
- $k_{13}$ and $k_{14}$ as inputs to the IT function.
- $x_1$ to $x_4$ as inputs to the IT function.
- $y_1$ to $y_4$ as outputs from the IT function.

FT Function:
- $k_{15}$ and $k_{16}$ as inputs to the FT function.
- $BP$ as an inverse function connected to $k_{15}$.
- $x_1$ to $x_4$ as inputs to the FT function.
- $y_1$ to $y_4$ as outputs from the FT function.

First AES Candidate Conference
Copyright NTT 1998
Design Rationale of IT / FT-Functions (cont.)

- **multiplication ⊗**
  - in order for each bit of the subkey to change many bits of output
  - four 32-bit integer multiplications

- **XOR ⊕**
  - improves the level of confusion by mixing incompatible group operations

- **byte permutation BP**
  - links different subblocks
IT-Function and FT-Function Overview

**IT Function:**
- $k_{13}$
- $k_{14}$
- $BP$

**FT Function:**
- $BP^{-1}$
- $k_{15}$
- $k_{16}$

Diagram showing the flow of inputs and outputs with $x_1$, $x_2$, $x_3$, and $x_4$ feeding into the IT function, and $y_1$, $y_2$, $y_3$, and $y_4$ coming out of the FT function.
Key Scheduling Part (1)
Key Scheduling Part (1)

Simple implementation
Key Scheduling Part (1)

Key setup time < 3-block encryption
Key Scheduling Part (1)

No simple relation
Key Scheduling Part (1)

All bits of master key equally influence all bits of subkeys
Key Scheduling Part (2)

Intermediate keys

<table>
<thead>
<tr>
<th>L_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>l_0</td>
</tr>
<tr>
<td>l_1</td>
</tr>
<tr>
<td>l_2</td>
</tr>
<tr>
<td>l_3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L_8</th>
</tr>
</thead>
<tbody>
<tr>
<td>l_28</td>
</tr>
<tr>
<td>l_29</td>
</tr>
<tr>
<td>l_30</td>
</tr>
<tr>
<td>l_31</td>
</tr>
</tbody>
</table>

Subkeys

k_1
k_2
k_3
k_4
k_16
k_15
k_14

Key Scheduling Part (2)

Intermediate keys

Deriving subkeys or master key from other subkeys is computationally infeasible
Key Scheduling Part (2)

Intermediate keys

Subkeys
Key Scheduling Part (2)

Intermediate keys

Subkeys

First AES Candidate Conference  Copyright NTT 1998
Key Scheduling Part (2)

Intermediate keys

Subkeys

First AES Candidate Conference

Copyright NTT 1998
Key Scheduling Part (2)

Deriving subkeys or master key from other subkeys is computationally infeasible.
Outline

- Overview
- Design
- Security
- Performance
- Conclusion
Security of Data Randomizing Part

- s-box is designed to provide reasonable security against
  - Differential cryptanalysis
  - Linear cryptanalysis
  - Higher order differential attack
  - Interpolation attack, etc.
## Properties of s-box

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value</th>
<th>Related Attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>bijectivity</td>
<td>OK</td>
<td>Differential/Linear</td>
</tr>
<tr>
<td>$w_H(a)$</td>
<td>$3 \leq w_H(a) \leq 5$</td>
<td>—</td>
</tr>
<tr>
<td>$w_H(b)$</td>
<td>$3 \leq w_H(b) \leq 5$</td>
<td>—</td>
</tr>
<tr>
<td>$p_s$</td>
<td>$2^{-4.67}$</td>
<td>Differential</td>
</tr>
<tr>
<td>$q_s$</td>
<td>$2^{-4.38}$</td>
<td>Linear</td>
</tr>
<tr>
<td>$r_s$</td>
<td>$2^{-2.59}$</td>
<td>(Differential-linear)</td>
</tr>
<tr>
<td>$\deg s$</td>
<td>7</td>
<td>Higher order differential</td>
</tr>
<tr>
<td>$\text{coeff}_{2^8} s$</td>
<td>254</td>
<td>Interpolation</td>
</tr>
<tr>
<td>$\text{coeff}_p s$</td>
<td>254</td>
<td>Interpolation</td>
</tr>
</tbody>
</table>

- $p$: prime, $256 < p < 512$
Security of Data Randomizing Part (cont.)

- s-box is designed to provide reasonable security against DC, LC, higher order differential attack, interpolation attack, etc.
- 9-round $E_2$ without $IT$ / $FT$-Functions has sufficient security against DC and LC
- $IT$ / $FT$-Functions are added for “insurance policy”
  - $E_2$ has 3-round margin + $IT$ / $FT$-Functions
Security of Key Scheduling Part

• No known weak keys
• No known equivalent keys
• No known complementation properties
Outline

• Overview
• Design
• Security
• Performance
• Conclusion
<table>
<thead>
<tr>
<th>Platform</th>
<th>Language</th>
<th>Key length (bits)</th>
<th>Key setup (clocks)</th>
<th>Encryption Decryption (clocks/block)</th>
<th>Encryption Decryption (bits/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Pentium Pro (200MHz)</td>
<td>ANSI C (Borland C++5.02)</td>
<td>128</td>
<td>2,076</td>
<td>711</td>
<td>36.0 M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>192</td>
<td>2,291</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>256</td>
<td>2,484</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assembly</td>
<td>all</td>
<td></td>
<td>420</td>
<td>61.0 M</td>
<td></td>
</tr>
<tr>
<td>Hitachi H8 / 300 (5MHz) 8bit CPU for smart card</td>
<td>Assembly</td>
<td>128</td>
<td>14,041</td>
<td>6,374</td>
<td>100.5 k</td>
</tr>
<tr>
<td></td>
<td></td>
<td>192</td>
<td>15,284</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>256</td>
<td>16,518</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEC 21164A (600MHz)</td>
<td>Assembly</td>
<td>all</td>
<td>600</td>
<td>128.0 M</td>
<td></td>
</tr>
</tbody>
</table>

_E2 requires no algorithm setup. The results contain no API overhead._
Current Hardware Performance

- CMOS 0.25 μm cell based library
- 1 Gbits/sec (typical)
- 482 Mbits/sec
- Total 127k gates
  - including key scheduling, control logic and buffers
- Not fully optimized
Outline

• Overview
• Design
• Security
• Performance
• Conclusion
Conclusion

$E_2$ is
Conclusion

$E2$ is

- Secure: secure against all known attacks with enough margin
Conclusion

$E2$ is

- Secure: secure against all known attacks with enough margin
- Fast: faster than DES
Conclusion

$E2$ is

- **Secure**: secure against all known attacks with enough margin
- **Fast**: faster than DES
- **Flexible**: efficient implementations on various platforms
E2 Home Page

http://info.isl.ntt.co.jp/e2/

Latest information is available.

e-mail: e2@isl.ntt.co.jp