Comparative Analysis of Robust, Fault Attack Resistant Architectures for Public and Private Cryptosystems

Konrad J. Kulikowski, Zhen Wang, Mark G. Karpovsky

Boston University

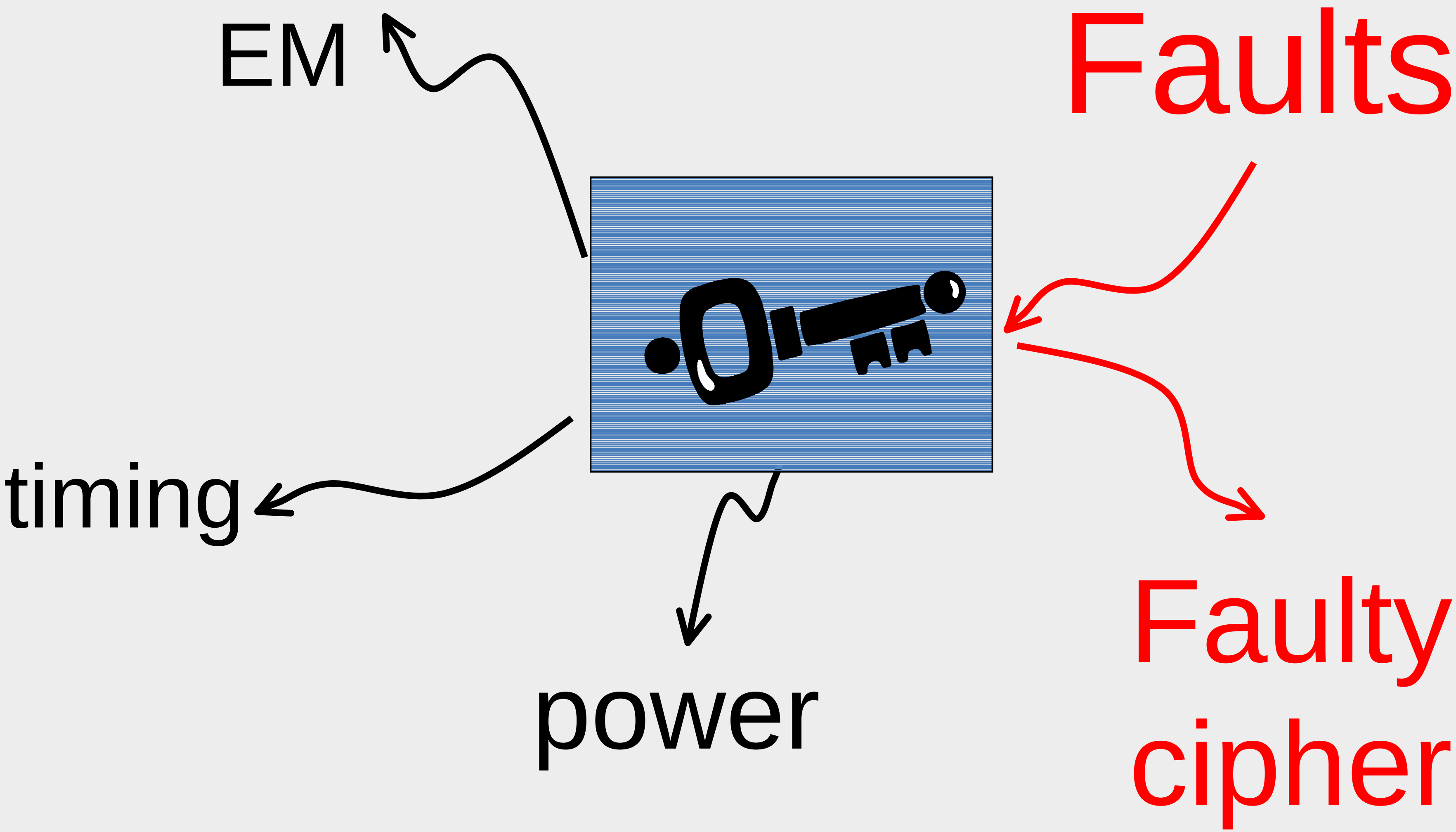
Reliable Computing Laboratory

FTDC, 8/10/2008

Outline

- Motivation for **robust** codes
- Robust code variants
  - Partially robust codes
  - Minimum distance robust codes
  - Minimum distance partially robust codes
- Constructions of codes
- **Applications**
- Case Study (AES)
- Case Study (Multipliers)

Side Channel Attacks



Error Model for Systematic codes

For a code $C = \{(x, y = f(x)\}$

Algebraic (binary) $x \in Z_2^k, y \in Z_2^r$

$$e = \tilde{w} \oplus w$$

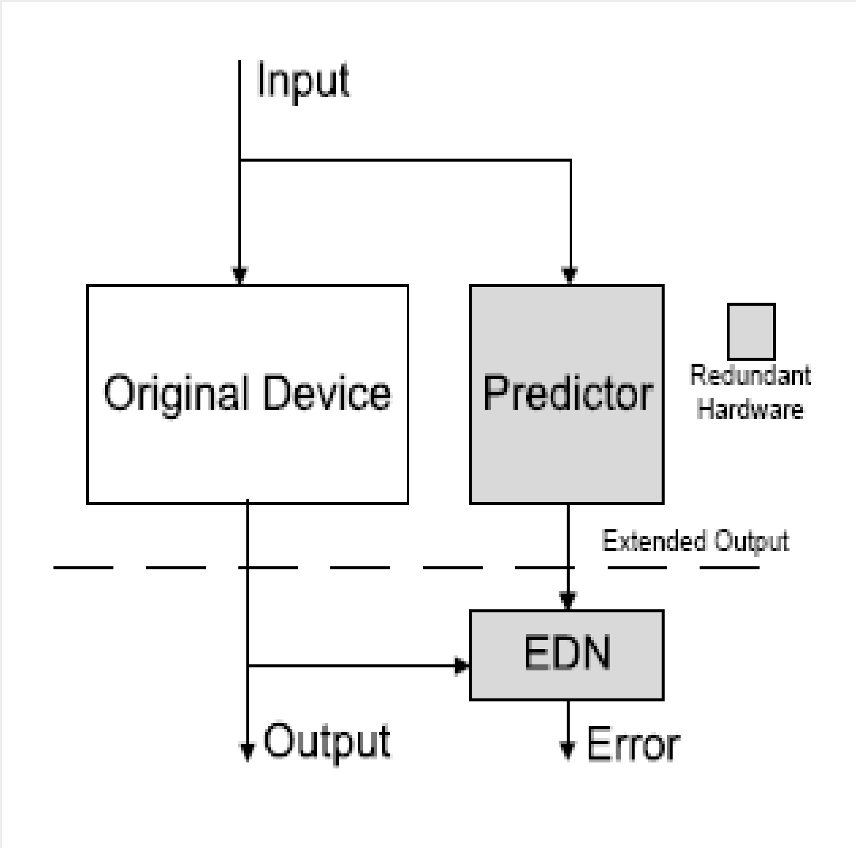
Arithmetic $x \in Z_{2^k}, y \in Z_p$

$$e = ((\tilde{x} - x) \bmod 2^k, (\tilde{y} - y) \bmod p)$$

For **linear** codes errors are missed iff

$$e \in C$$

General Architecture



Robust Codes Characteristics

- **Equal Protection for all errors**
  – Unpredictable attacker
- Predictable worst case performance
- **All** errors are detectable

Robust Codes Characteristics

- Detection improves as more messages are distorted

- Detectability of any given error depends

on the message (cipher) which **cannot** be predicted by the attacker**.**

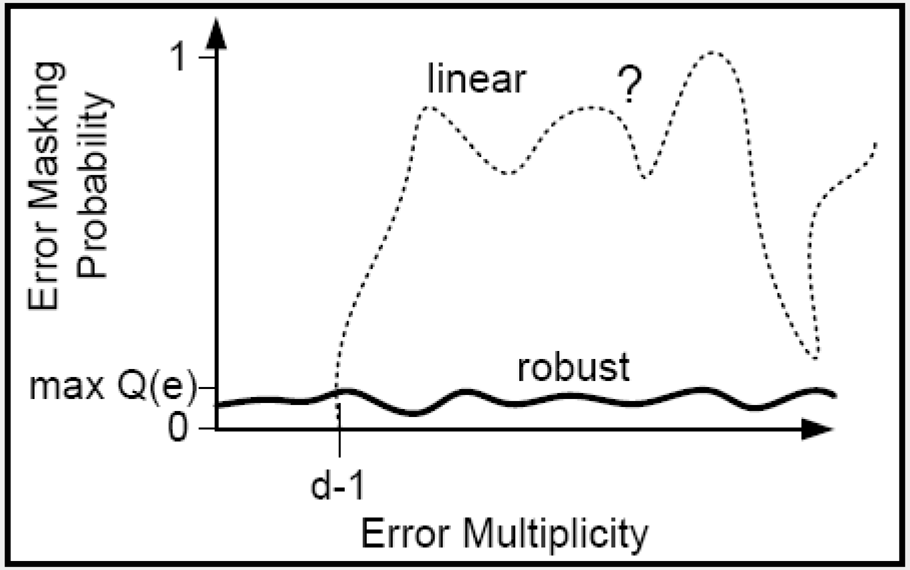
Robust Codes Definition

Kernel of a code (set of undetectable errors):

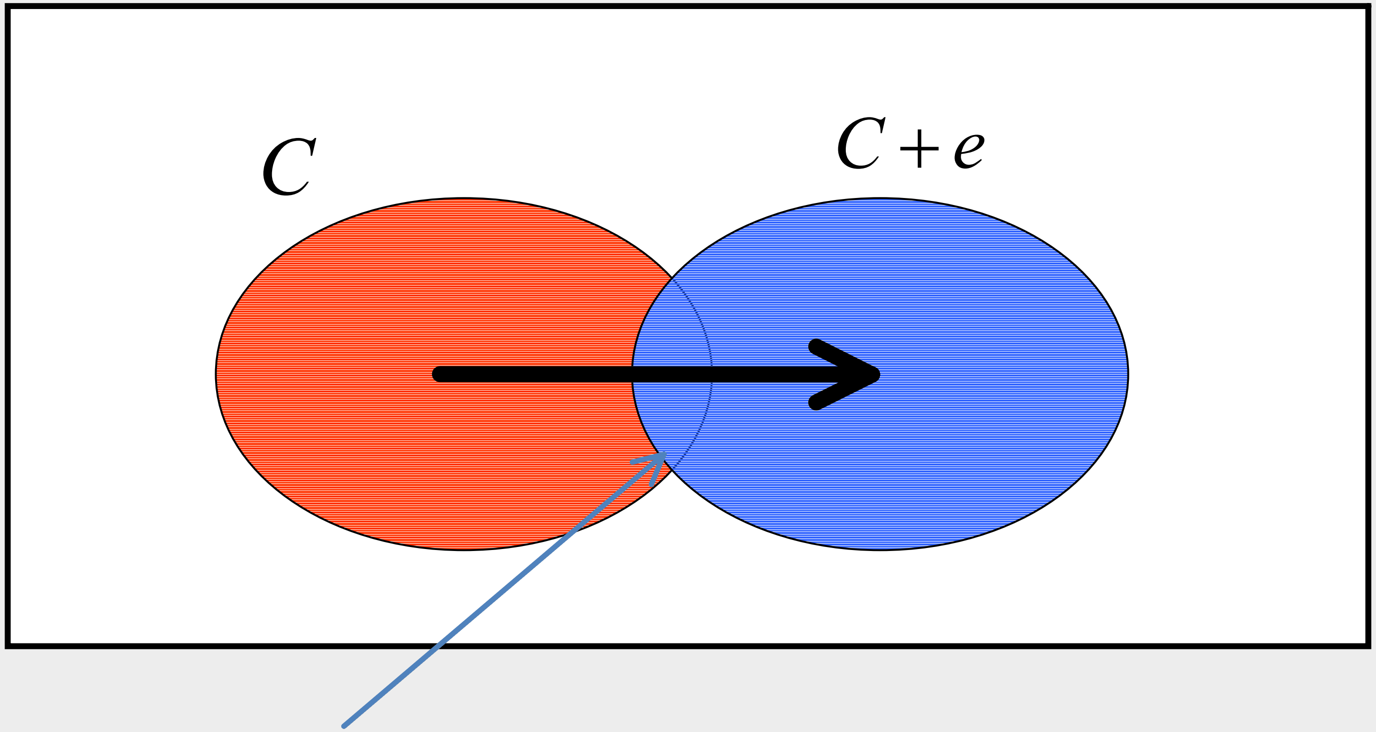
$$K = \{e \mid e + w \in C \; if \; w \in C\}$$

Error masking probability:

$$Q(e) = \frac{|\{w \mid w \in C, w + e \in C\}|}{|C|}$$



Robust Error Detecting Codes



$$R = \max | C \cap (C + e) | < | C |$$

Every error is missed for at most R messages (max Q(e)=R/|C|)
Detection probability increases as more erroneous messages are observed

Previous Work

- Kulikowski K, M.G. Karpovsky. A. Taubin. ***Robust Codes and Robust, Fault Tolerant Architectures of the Advanced Encryption Standard****.* Journal of Systems Architecture special issue on Embedded Cryptographic Hardware. vol. 53, pp. 138-139, 2007.
- Kulikowski K, M. G. Karpovsky, A. Taubin. ***Fault Attack Resistant Cryptographic Hardware with Uniform Error Detection.*** FDTC, 2006.
- Gaubatz, G., B.Sunar, M.G.Karpovsky, ***Robust Residue Codes for Fault-Tolerant Public-Key Arithmetic.*** FDTC ,2006
- Karpovsky, M.G., K. Kulikowski, A. Taubin, ***Robust Protection Against Fault-Injection Attacks of Smart Cards Implementing the Advanced Encryption Standard****.* DSN'04, 2004.
- Mark Karpovsky and Alexander Taubin, ***A New Class of Nonlinear Systematic Error Detecting Codes***, IEEE Trans Info Theory, Vol 50, No.8, 2004, pp.1818-1820

Robust Codes:

- Robust

$$|K| = 0, Q(e) < 1$$

- Partially Robust (FDTC'06, DSN'04, CARDIS'04)

$$|K| < 2^k, Q(e) < 1 \; if \; e \notin K$$

- Minimum Distance Robust

$$|K| = 0, Q(e) = 0 \; if \; \|e\| < d, \;\; Q(e) < 1$$

- **Minimum Distance Partially Robust**

$$|K| < 2^k, Q(e) = 0 \; if \; \|e\| < d, Q(e) < 1, e \notin K$$

Fully robust codes

$$C = \{(x, f(x)) \mid x \in Z_2^{\;k}\}$$

f(x) "highly nonlinear function"

optimum when *f(x)* is a "perfect nonlinear function"

$$f(x = (x_0, x_1, \ldots, x_{k-1})) = x_0 x_1 \oplus x_2 x_3 \oplus \ldots \oplus x_{k-2} x_{k-1}$$

k	*r*	*d*	*|K|*	*maxQ(e)*
32	1	1	0	0.5

Fully robust codes (arithmetic)

$$C = \{(x, f(x)) \mid x \in Z_{2^k}\}$$

$$f : Z_{2^k} \rightarrow Z_p$$

$$f(x = (x_0, x_1, \ldots, x_{k-1})) = x_0x_1 + x_2x_3 + \ldots + x_{k-2}x_{k-1}, \quad \bmod p,$$

$$x_i \in Z_r, \; r = \lceil \log_2 p \rceil \quad , \quad p = 2^{16} - 15$$

k	*r*	*d*	*maxQ(e)<*
64	*16*	*1*	2^{-15}

Minimum Distance Robust Codes

$$C = \{(x, \pi(x), f(x)) \mid x \in Z_2^{\,k}\}$$

$\{(x,\pi(x))\}$ is a linear code with distance *d*

f(x) is a perfect nonlinear function

$$\pi(x = (x_0, x_1, \ldots, x_{k-1})) = x_0 \oplus x_1 \oplus \ldots \oplus x_{k-1}$$
$$f(x = (x_0, x_1, \ldots, x_{k-1})) = x_0 x_1 \oplus x_2 x_3 \oplus \ldots \oplus x_{k-2} x_{k-1}$$

k	*r*	*d*	*|K|*	*maxQ(e)*
32	2	2	*0*	0.5

Partially Robust Codes

$$C = \{(x, f(l(x))) \mid x \in Z_2^{k}\}$$

$\{(x, l(x))\}$ is a linear code

f(x) is a nonlinear function

$l(x) =$ encoding function of (38,32) Hamming

$f(l(x)) = (l(x))^3 \quad f : Z_2^6 \rightarrow Z_2^6$

k	*r*	*d*	*|K|*	*maxQ(e)*
32	6	1	2^{26}	2^{-5}

Partially Robust (arithmetic)

$$C = \{(x, f(l(x)) \mid x \in Z_2^k\}$$

$$l : Z_2^k \to Z_p \qquad f : Z_p \to Z_p$$

$$l(x) = x, \quad \text{mod } p$$

$$f(l(x)) = (l(x))^2 \text{ mod } p, \quad f : Z_p \to Z_p$$

e is "bad" if $Q(e) > 0.5$

k	r	d	Prob. of "bad"
64	16	1	$\approx 2^{-31}$

Minimum distance partially robust

Shortened **Vasil'ev*** constructions

$$C = \{(u, (u,0) \oplus v, p(u) \oplus f(v))\}$$

$u \in Z_2^a, v \in V$ $\quad V$ is a shortened Hamming code of length $m \geq a$

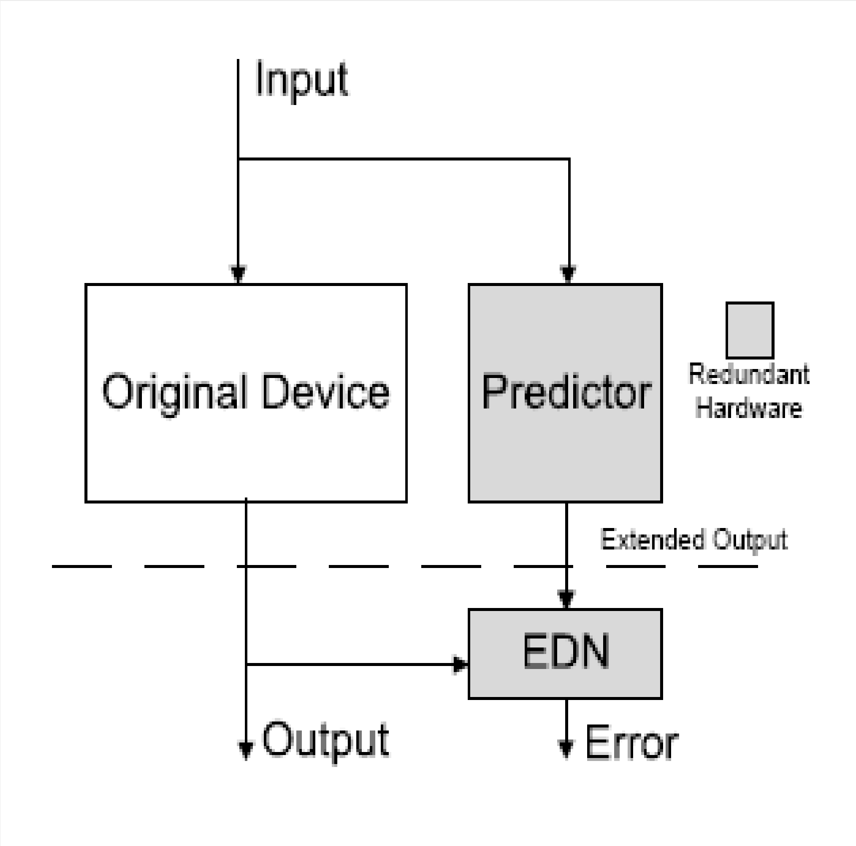
$p(u)$ **Linear Parity Function**

$f(v)$ **Nonlinear Funtion** $\quad f : V \to Z_2$

k	*r*	*d*	*|K|*	*maxQ(e)*
32	6	3	2^a	0.5

*J. L. Vasil'ev. **On nongroup close-packed codes**. In *Probl.Kibernet.*, volume 8, pages 375–378, 1962.

Case Study (AES)



217 XOR gates

Linear sub-block of AES
32-bit input
32-bit output (k=32)

Protected with
linear parity, *r=1*
robust party, *r=1*
min dist robust, *r=2*

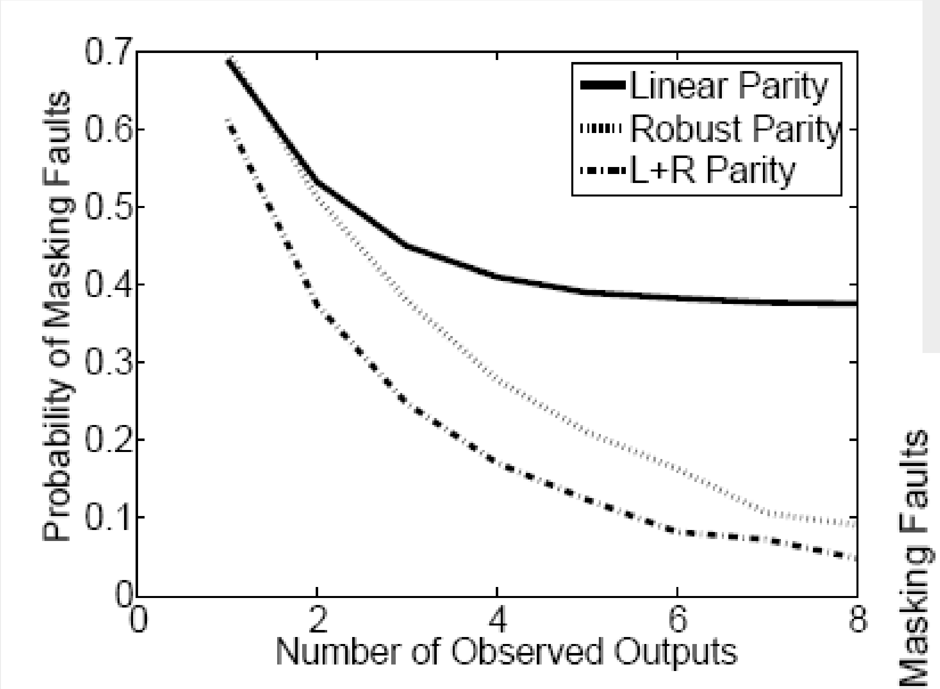
Hamming, *r=6*
Vasil'ev, *r=6*
partially robust, *r=6*

Case Study (AES)

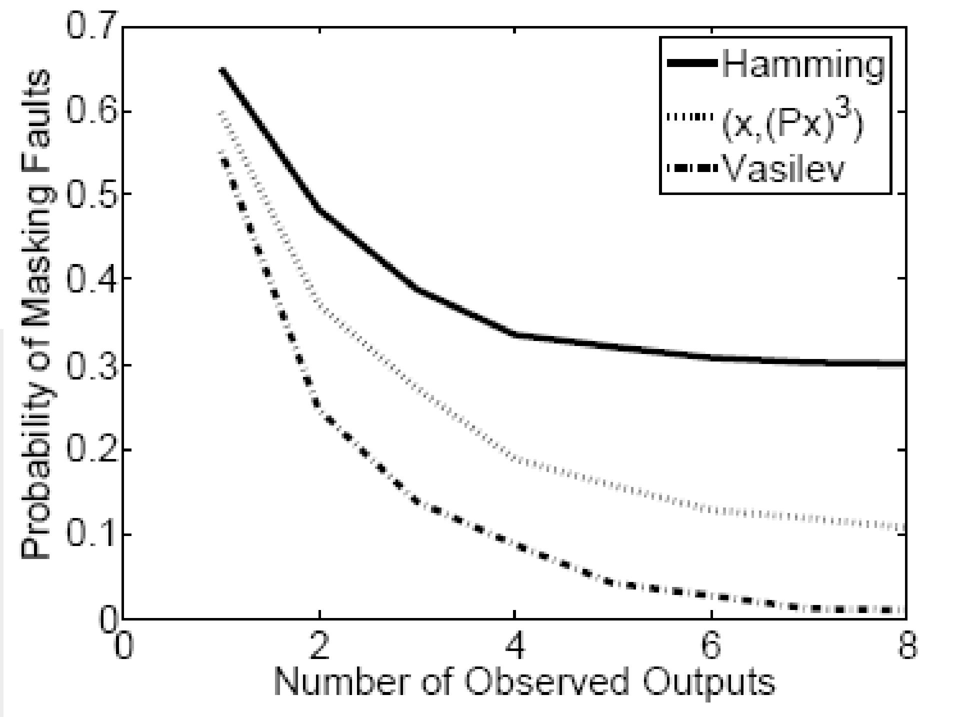
k=32

	r	Predictor (gates)	EDN (gates)	Overhead (%)	|K|	max Q(e)
Linear parity	1	31	32	30%	2^{32}	1
Robust parity	1	185	32	100%	0	2^{-1}
Linear and robust parity (d=2)	2	196	64	120%	0	2^{-1}
Hamming	6	253	80	153%	2^{32}	1
Vasil'ev	6	292	116	188%	2^{6}	2^{-1}
Partially Robust	6	432	266	322%	2^{26}	2^{-6}

AES, Single Stuck-at Faults

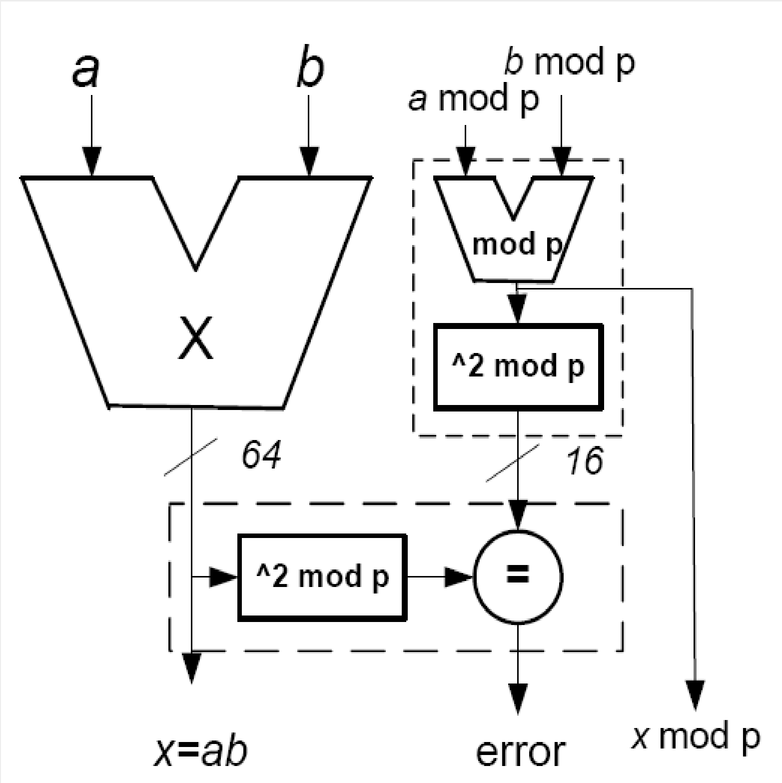


Robustness reduces error masking of faults

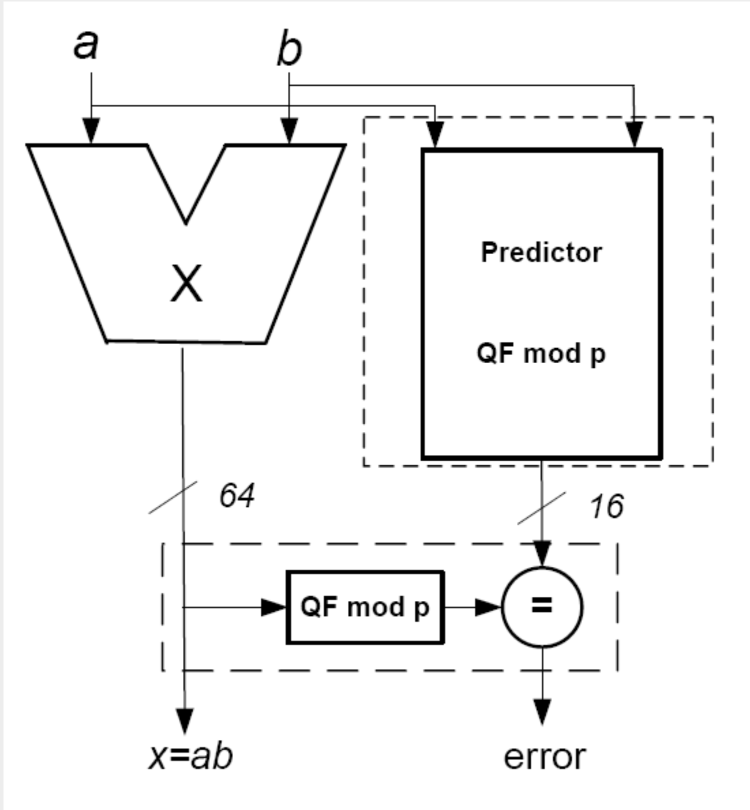


Case Study (**Multipliers**)

32-bit multiplier protected with arithmetic robust codes

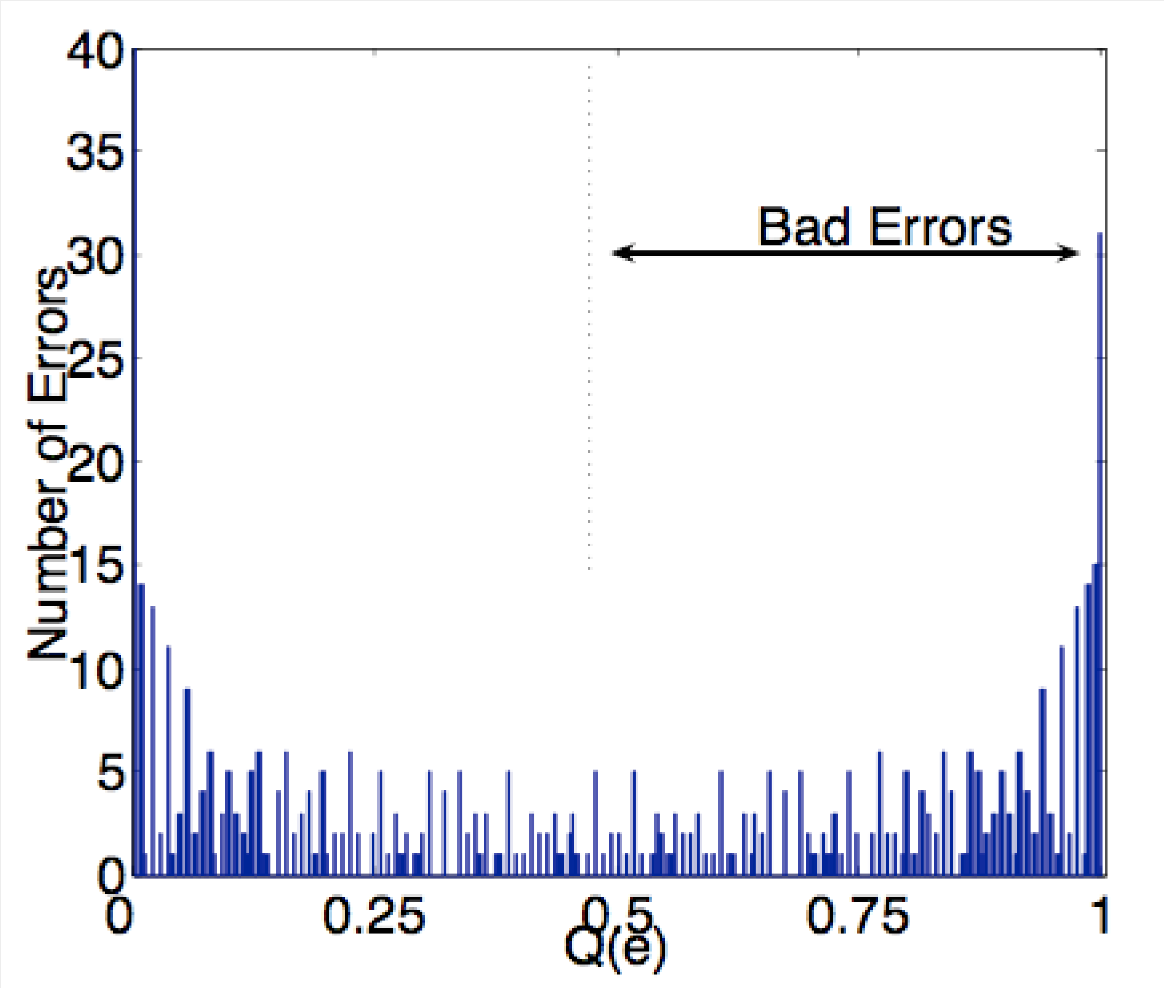


Partially robust, k=64, r=16

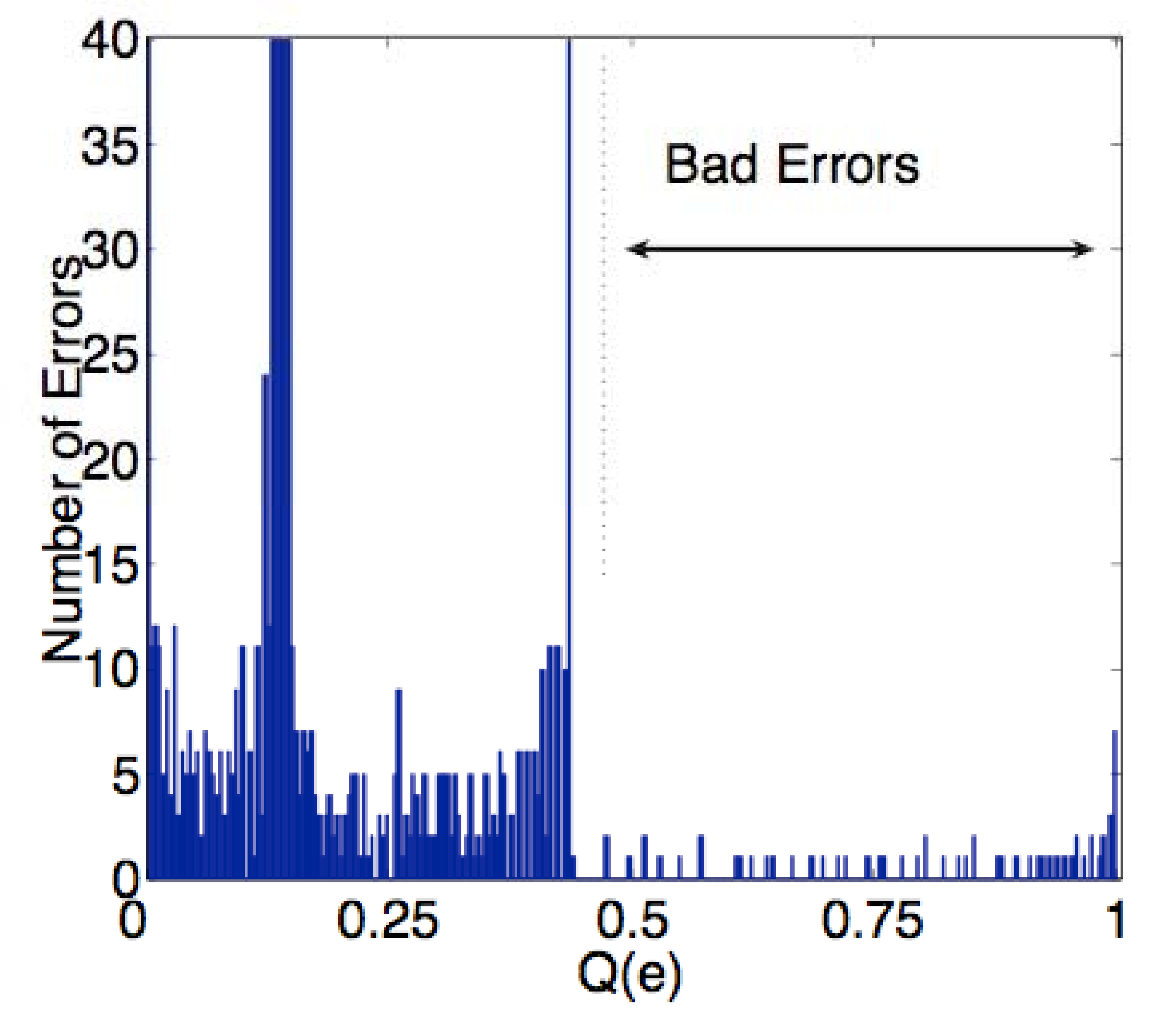


Robust, k=64, r=16

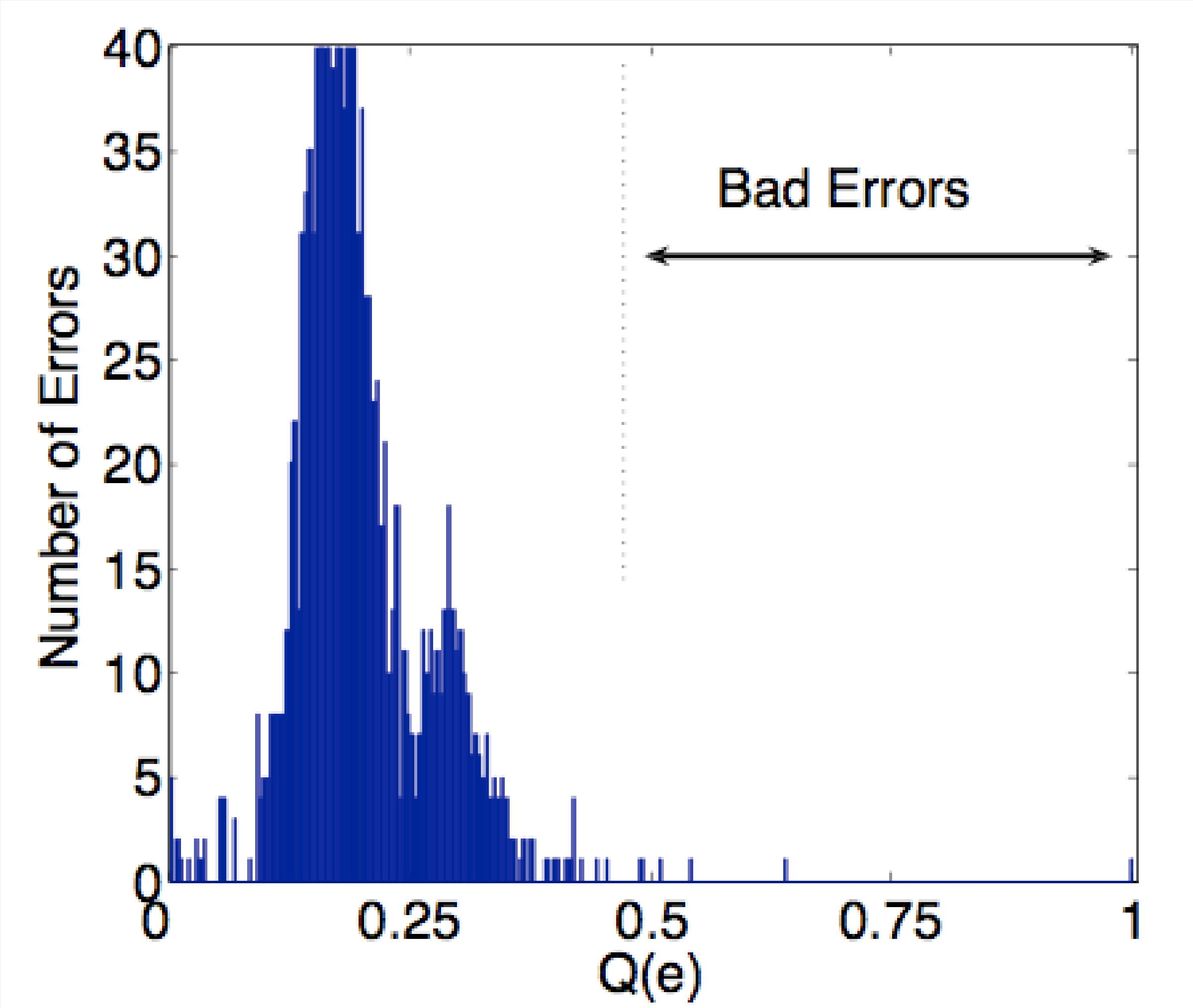
Linear AN-codes (Multiplier, k=8,p=5)



Partially Robust (**Multiplier**,k=8,p=5)



Robust (Multiplier, k=8,p=5)



Codes for Multipliers (Summary)

k=64

	Linear	Partially Robust	Robust
r	16	16	16
Overhead (%)	25%	75%	200%
Probability of bad error	$\approx 2^{-15}$	$\approx 2^{-31}$	0

Summary

- **Fully robust** codes have a high overhead, but detect all errors
- Overhead can be reduced while maintaining many of the robust properties by using **partially robust** codes
- **Minimum distance robust** codes beneficial for detection of natural errors