EC500
Design of Secure and Reliable Hardware

Lecture 1 & 2

Mark Karpovsky
January 17th, 2013

Security
Errors injected by the attacker (active attacks)

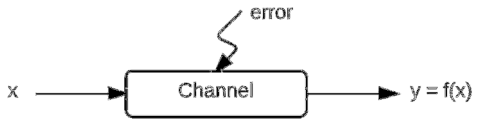
Reliability
Errors injected by random sources e.g. variation in power supply, temperature, α particle, aging, … (passive attacks)

Design for Security
Goal: Device detects that it is under attack and disables itself

Design for Reliability
Goal: Device detects, locates, or corrects random errors

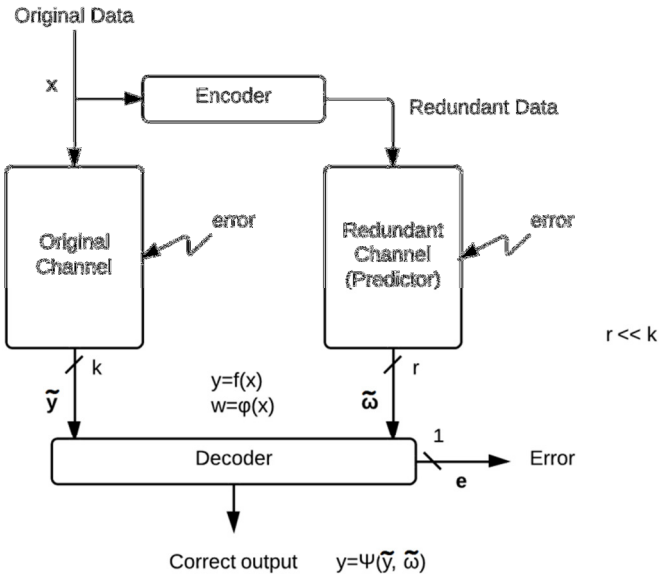
Main Tool: Redundancy
1. Information redundancy
2. Space redundancy
3. Time redundancy

Channel



$y = x$ ↔ Communication e.g. memory
$y = f(x)$ ↔ Computation

Block Diagram



Symmetric Cipher Private Key System

Example

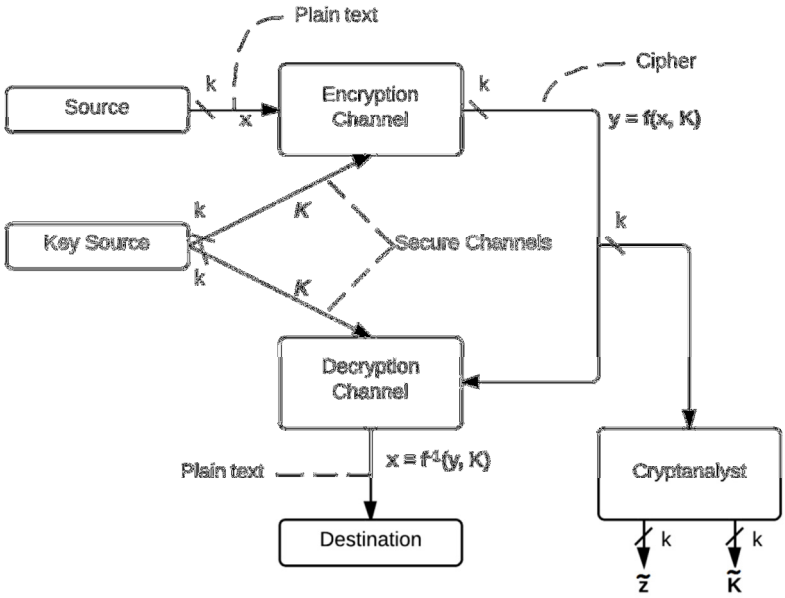
AES – 128/256 bits for plain text and ciphers

$k = 128/256$

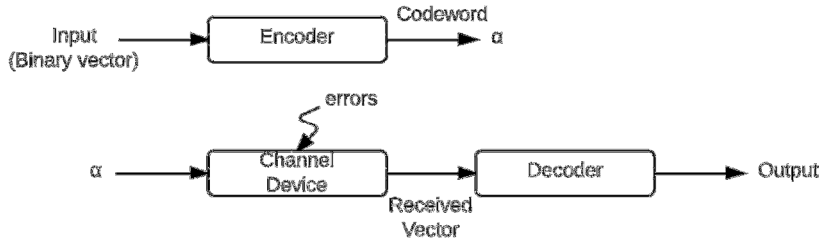
x – message plain text	k-bits
y – cipher	k-bits
K – key (secret)	k-bits

(Secret key may be distributed between users!)

Model of Conventional Cryptosystem



Mathematical Theory of Detection, Location, and Correction of Errors in Computers and Communication Channels



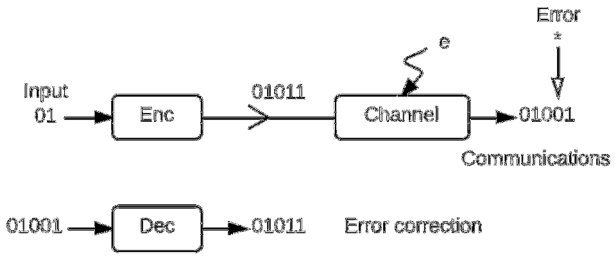
Example (Encoder ENC)

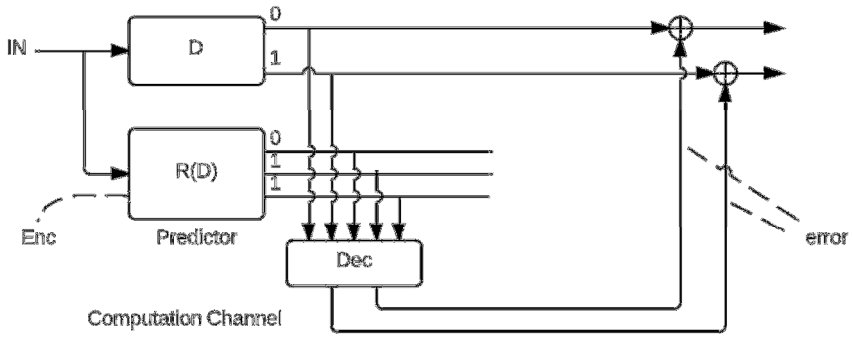
$00 \rightarrow 00000$

$01 \rightarrow 01011$ ← Error correction

$10 \rightarrow 10101$

$11 \rightarrow 11110$





Attacks (Fault Injection)

Goals:

1. Substitute Data
2. Jamming
3. Obtain a sensitive data (pwd, key) by comparing fault-free and faulty outputs

Error:

$y \rightarrow y \oplus e = \tilde{y}$

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

Assumption:

The attacker knows:

1. Function $y = f(x)$ (Channel function)
2. Function $\omega = \varphi(x)$ (Encoding function)
3. Function $y = \Psi(\tilde{y}, \tilde{\omega})$ (Decoding function)
4. Error function $e(\tilde{y}, \tilde{\omega}) \in \{0,1\}$

Weak Attack

- Attacker can inject any error e
- Fault-free outputs y are uniformly distributed (and not known to the attacker)
- Fault-free outputs $y = f(x, k)$ may also depend on secret key K

Strong Attack

- Attacker can inject any error e for any given output $y = f(x, K)$

Main tool for design of reliable and secure hardware → Error detecting/locating/correcting codes.

Mathematical Tools:

1. Finite fields
2. Linear spaces over finite fields

Examples of Codes:

1. Repetition
2. Parity

Design Criteria

1. Performance → Probability of detection
2. Space overhead → Area, gate count
3. Time overhead → Latency, delay
4. Power overhead

Security is more expensive than reliability.

Problem: How to distinguish between random errors (which can be detected and/or corrected) and attacks (when the channel should be disabled).

Consider code $C_1 = \{00000, 01011, 10101, 11110\}$ (k=2, r=3). Assume that all four messages are equiprobable.

If $e = 00001$ (last bit is distorted), then the probability of missing $Q(00001) = 0$. Let $\|e\|$ – number of nonzero components in error vector e. Then for C_1, if $0 < \|e\| \leq 2 \rightarrow Q(e) = 0$. But if $e = 01011$ then $Q(e) = 1$! Thus if only errors e may appear in the channel are errors with $\|e\| \leq 2$, then the code is good (case of random errors).

But if errors are injected by an attacker (weak attack), the same code is very bad.

- Code C can <u>detect</u> l errors if any two vectors from the code have at least $l+1$ different components. If $0 < \|e\| \leq l$, then $Q(e) = 0$, i.e. $\|a \oplus b\| \geq l$ if $a, b \in C$.
  Hamming distance between any two codewords in C is at least $l+1$ or $d(C) \geq l+1$
- Code C can <u>correct</u> l errors if $d(C) \geq 2l+1$
- Code C is <u>linear</u> if $\emptyset = (0 \dots 0) \in C$ and for any $a, b \in C$, $a \oplus b \in C$
- Code C_1 from the previous example is linear
  For linear codes: $Q(e) = \begin{cases} 0, e \notin C \\ 1, e \in C \end{cases}$
- If an error is a codeword, it is never detected by linear codes.
- Linear codes are bad even for weak attacks

We define a "ball" of radius l around center y as $B_y = \{\tilde{y} | d(\tilde{y}, y) \leq l\}$ and we can define the "volume" of a ball as the number of codewords contained in the ball to be equal to $V = 1 + \sum_{i=1}^{l} \binom{n}{i} (q-1)^i$

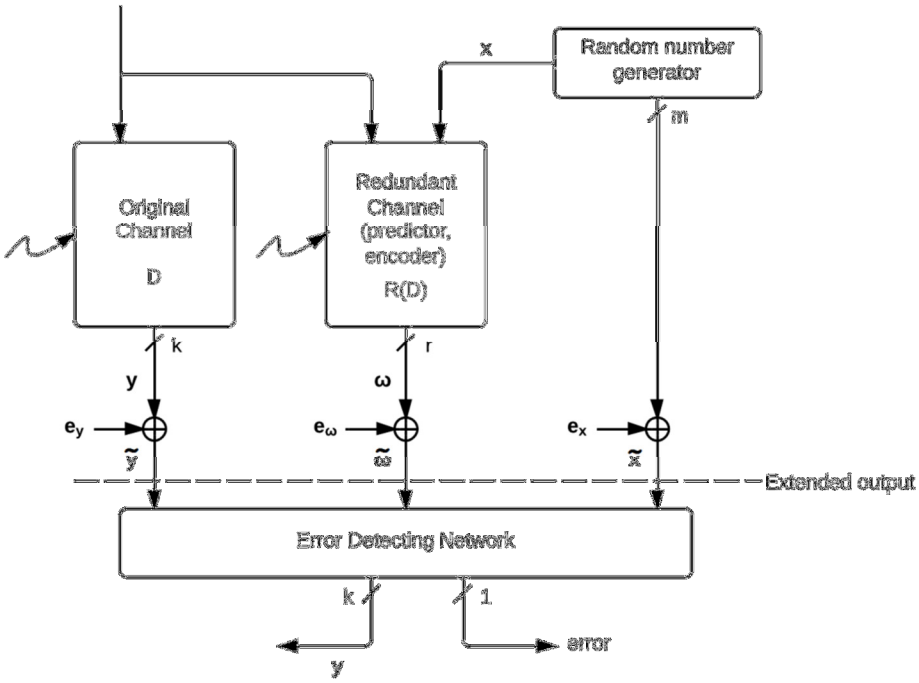
For example, for a binary code with length 4, we can calculate the volume of the ball with $l = 2$ as $1 + 4 + 6 = 11$

Consider code $C_2 = \{000,010,100,111\}$ (k=2, r=1).
If $e = e_1 = 001 \rightarrow Q(e_1) = 0$
But if $e = e_2 = 010 \rightarrow Q(e_2) = 0.5$

e_2 is not detected for messages 000 and 010
e_2 is detected for messages 100 and 111

For C_2 for any e, $Q(e) \leq 0.5$. C_2 is better than C_1 for weak attacks. If the channel is <u>lazy</u> (error repeat for different messages), then for C_2 after t clocks, $Q \leq 0.5^t$ – probability of missing error e.

Secure Architecture for Strong Attacks



Summary on Attacks

	Message y	Error e	Application
Passive Attacks	R	R	Reliability
Weak Active Attack	R	A	Security (Weak)
Strong Active Attack	A	A	Security (Strong)

R – random

A – controlled by the attacker