

STRATEGY OF SECURITY

Adhering Security with Performance



under the supervision of

Prof. Debdeep Mukhopadhyay
Prof. Rajat Subhra Chakraborty

Souvik Sonar
SEAL,CSE, IIT Kharagpur

MOTIVATION^[1]

SECURITY CHALLENGES

- Advanced Cryptanalysis
- Side Channel Attacks ^{[2][3]}
- Fault Attacks ^{[4][5]}
- Physical Abrasion, Chemical Etching
- Focussed Ion Beam Technology

PERFORMANCE ISSUES

- Speed of Computation
- Power Consumption
- Availability
- Complexity
- Stability & Durability

STRATEGY

COUNTERMEASURES^[1]

- Sensor Information
- Detection of Errors (Spatial, Temporal, Information)
- Detection of software modification
- Randomization in instruction order
- Addition of dummy operations
- Masking internal computations
- Correlation between physical values and data processed
- Modification of functional behaviour of the circuit

OVERVIEW

- Define a Host System : Conditional Access System (CAS) for Smart Card
- Description of various Protections and Countermeasures
- Impact on the Performances over Security Parameters (Theoretical Analysis)
- Smart Dynamic Management between attacks and normal use cases (Fuzzy Approach)
- Hardware / Software prototype
- Simulation Scenarios
- Limitations and Challenges
- Conclusion
- References

SMART CARD FOR CAS

JAVACARD APPLICATION (App)

- Criteria to be met to grant access to the content (Radio, PayTV)
- Smart Card ^[6] (Sensitive Information) => Receiver for deciphering the content
- 1st level => Management Key (MK) => Access Rights (subscriptions, validity) & EK
- 2nd level => Exploitation Key (EK) => Access Criteria (CA) to content & CW
- EK is changed approximately once in every month.
- 3rd level => Control Word (CW) => protects the content
- CW is changed approximately once in every 5 to 10 secs.

Sensitivity of the Data , 'DS' = {0,1,.....,5}

VIRTUAL MACHINE (VM)

- Java Card 2.2.2^[7] for Compilation
- Global Platform Standard^[8] for Management Capabilities
- Efficient Software Security Features
- Installation of On-Card Applications obeying above standards

MICRO-CONTROLLER (HW)

- 5-Stage Pipelined 32-bit Harvard RISC Micro-controller
- Instruction memory : 640kB ROM
- Data memory : 256kB RAM & 128kB EEPROM
- 2-UARTs peripherals (ISO7816 & RS232)
- AES CryptoEngine

PROTECTIONS & COUNTERMEASURES

SECURITY SENSORS^[1]

Name	Values	Description	Updated by
<i>LS</i>	{0, 1, ..., 5}	# of triggers of the light sensor	HW
<i>VS</i>	{0, 1, ..., 10}	# of triggers of the voltage sensor	HW
<i>EFE</i>	{0, 1, ..., 10}	# of corrupted execution flow	VM
<i>CE</i>	{0, 1, ..., 10}	# of corrupted execution	VM
<i>PE</i>	{0, 1, ..., 10}	# of wrong PIN	App
<i>NE</i>	{0, 1, ..., 10 ³ }	# of methods processed without error	VM
<i>ME</i>	{0, 1, ..., 10 ⁴ }	# of MAC errors	App
<i>CO</i>	{0, 1, ..., 10 ⁷ }	# of cryptographic execution	App

REDUNDANCY LEVEL , RL

- Countermeasure for Fault Attack
- Detection of Errors by performing same computations several times
- Comparing the results => *If* same Error Free *Else* CE is incremented.
- *If* redundancy countermeasure is not activated , RL = 1
Else host has fault-tolerant capabilities (generally RL >= 3)

PROTECTIONS & COUNTERMEASURES

INSERTION OF DUMMY INSTRUCTIONS

- Execution of Program = Execution of **D** useful instructions
+ Execution of **N** dummy instructions
- D and N are the random variables for useful and dummy instructions^[9]
- Domain of $D = \{1;2;\dots;\mathbf{D}\}$
- Domain of $N = \{0;2;\dots;\mathbf{N}\}$ $N = 0$ implies no countermeasure
- D and N follow uniform distributions.

RANDOM POWER GENERATORS

- Blur the power consumption => Random Number Generators (RNG)
- $x(t)$ = Power Consumption at each step obeying Gaussian (or Normal) *pdf* with mean = $\mu_c(t)$ and constant standard deviation = σ_c
- RNG's = R (Identical) each with mean = μ_R and constant standard deviation = σ_c
- Power Consumptions of R are statistically independent.

$$pdf(x_{tot}(t)) = \frac{e^{-\frac{(x_{tot}(t) - \mu_c(t) - R \cdot \mu_R)^2}{2 \cdot \sigma_c^2 \cdot (1 + R^2)}}}{\sigma_c \cdot \sqrt{1 + R^2} \cdot \sqrt{2 \cdot \pi}}$$

$R = 0$ implies
no countermeasure

QUANTIFICATION FOR IMPACT ANALYSIS^[1]

- ***FSCA*** : Ratio between the number of curves needed for adversary when the countermeasure is activated and the number of curves without countermeasures
- ***FDFA*** : Ratio between the number of experiments require for adversary when the countermeasure is activated and the number of experiments without countermeasures
- ***Ftime*** : Ratio between the duration of a computation with the countermeasure and the duration of the same computation without countermeasure
- ***FNRJ*** : Ratio between the energy consumption with the countermeasure and the energy consumption without countermeasure

FSCA = Gain in terms of SCA (Higher)

FDFA = Gain in terms of DFA (Higher)

Ftime = Loss in terms of speed (Lower)

FNRJ = Loss in terms of energy (Lower)

IMPACT OF REDUNDANCY

- **FSCA** : Redundant computations generate RL identical power traces that could be advantageous to adversary.^[12]
- **FDFFA** : The adversary have to avoid the update of CE and realize several faults of the same value, noted e_0 during RL computations and mount the attack on say q -bits. If the faults are equally probable, then probability of realizing the same fault e_0 during RL computations = $(1/2^q)^{RL-1}$.
- **Ftime** : We assume redundant computations are not performed in parallel and comparison of results are negligible. Redundancy countermeasures increases the computation time by factor RL.
- **FNRJ** : Energy consumption for comparison of results are negligible. Redundancy countermeasures increases the energy consumption by factor RL.

$$FSCA_{RL} = \frac{1}{RL}$$

$$FDFFA_{RL} = (1/2^q)^{RL-1}$$

$$FTime_{RL} = RL$$

$$FNRJ_{RL} = RL$$

IMPACT OF INSERTION OF DUMMY INSTRUCTIONS

Let m^{th} valid instruction set computes the result.

Each instruction is executed in one clock cycle (equal interval).

$x \in X = \text{Random Variable equal to the number of clock cycle associated with execution of } m.$

$$x = \sum_{i=1}^k (d_i) + \sum_{i=1}^k (n_i) \{ k \mid \sum_{i=1}^k (d_i) = m \}$$

$$x = m + \sum_{i=1}^k (n_i)$$

We consider $m \gg D$, $x \sim m + \sum_{i=1}^q (n_i)$ where $q = 2m/(D+1)$

Under this condition, X follows normal distribution with (μ_X, σ_X)

$$\mu_X = m + q \cdot \mu_N = m \cdot (1 + N/2)$$

$$\sigma_X^2 = q \cdot \sigma_N^2 = m \cdot \frac{N \cdot (N + 2)}{6 \cdot (D + 1)}$$

For instance, m is chosen to obey uniform distribution between $m - \sigma_X$ and $m + \sigma_X$ with equally probability $1/2\sigma_X$ [10]

IMPACT OF INSERTION OF DUMMY INSTRUCTIONS

- **FSCA** : Number of curves necessary for the adversary to attack is $2\sigma_x$ ^[10]
- **F DFA** : We suppose the attacker is able to target clock cycles comprising between $m - \sigma_x$ and $m + \sigma_x$. He has only one chance out of $2\sigma_x$ to modify instruction m .
- **Ftime** : Computation time is increased by factor $(1+N/2)$ as $\mu_x = m \cdot (1+N/2)$
- **FNRJ** : Power consumption D and N are same so energy consumption is also increased by factor $(1+N/2)$

$$FSCA_{IDI} = \begin{cases} 1 & \text{if } N=0 \\ 2 \cdot \sqrt{m \cdot \frac{N \cdot (N+2)}{6 \cdot (D+1)}} & \text{otherwise} \end{cases}$$

$$F DFA_{IDI} = FSCA_{IDI}$$

$$FTime_{IDI} = 1 + N/2$$

$$FNRJ_{IDI} = FTime_{IDI}$$

IMPACT OF RANDOM POWER GENERATOR

- **FSCA** : δ = Amplitude of difference of side-channel properties
 σ_c = Standard Deviation of the curve
Number of curves necessary for the adversary to attack is greater than $(\sigma_c/\delta)^2$ [10]
- **F DFA** : This mechanism does not protect against Fault Attacks supposedly
- **Ftime** : Computation time is not increased by activation of this mechanism
- **FNRJ** : Power consumption of an RNG is directly proportional to the mean, $\mu_c(t)$

$$FSCA_{RPG} = 1 + R^2$$

$$F DFA_{RPG} = 1$$

$$FTime_{RPG} = 1$$

$$FNRJ_{RPG} = (1 + \alpha \cdot R)$$

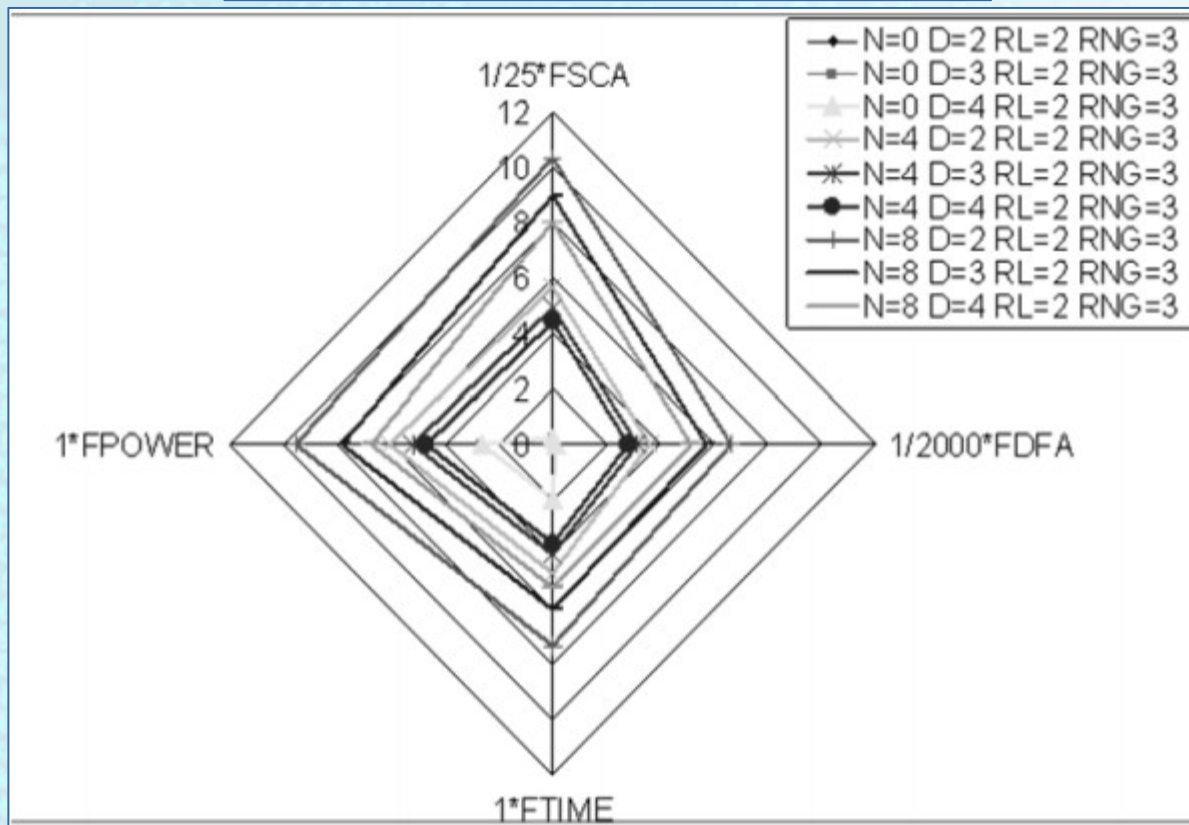
COMBINATION OF COUNTERMEASURE [1]

$$FSCA = FSCA_{RPG} \cdot FSCA_{IDI} \cdot FSCA_{RL}$$

$$FDFA = FDFARPG \cdot FDFARL^{-1} \cdot FDFARL$$

$$FTime = FSpeed_{RPG} \cdot FSpeed_{IDI} \cdot FSpeed_{RL}$$

$$FNRJ = FNRJ_{RPG} \cdot FNRJ_{IDI} \cdot FNRJ_{RL}$$



DESIGN DIFFICULTIES SMART CARD

- Security Level should be persistent for several years.
- Performance Level should be high
- Availability has to be high. Should be resistance to “anomalous” conditions as well.
- Has to deal and process data with various sensitivity levels.
- Power Consumptions must be low for embedded environment.
- Has to be inexpensive

STRATEGY OF SECURITY^[1]

- Dynamically modify the setups of the countermeasures
- Switch from high performance state to low secured state and vice-versa.
- Distinguish between anomalies and attacks.

E.g. -

Case 1: Voltage Sensor threshold is low

Gets triggered even when it is connected to low quality card reader
interpreting fault attack at application level

Anomaly considered as Attack => “False Positive Case”

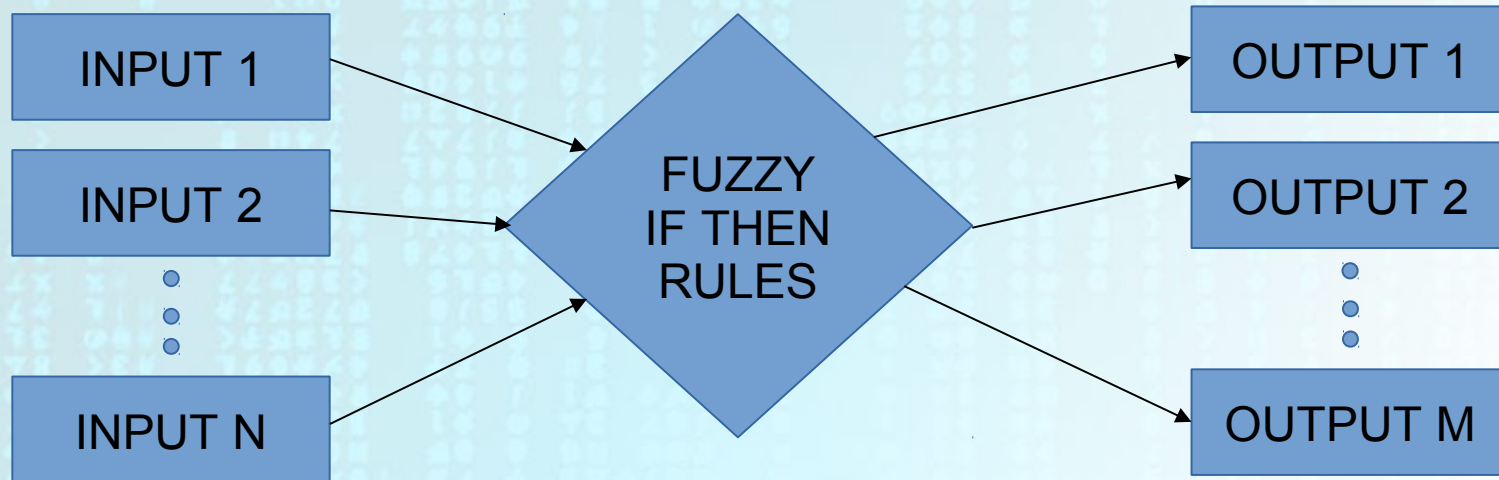
Case 2: Voltage Sensor threshold is high

Real glitch attacks may not be detected.

Attack considered as Normal => “False Negative Case”

- Approach decomposed into three different processes
 1. Information about state of the host system
 2. Computing attack levels and anomaly levels
 3. Modifying the parameters of the countermeasures

FUZZY LOGIC : AN INTRODUCTION^[11]

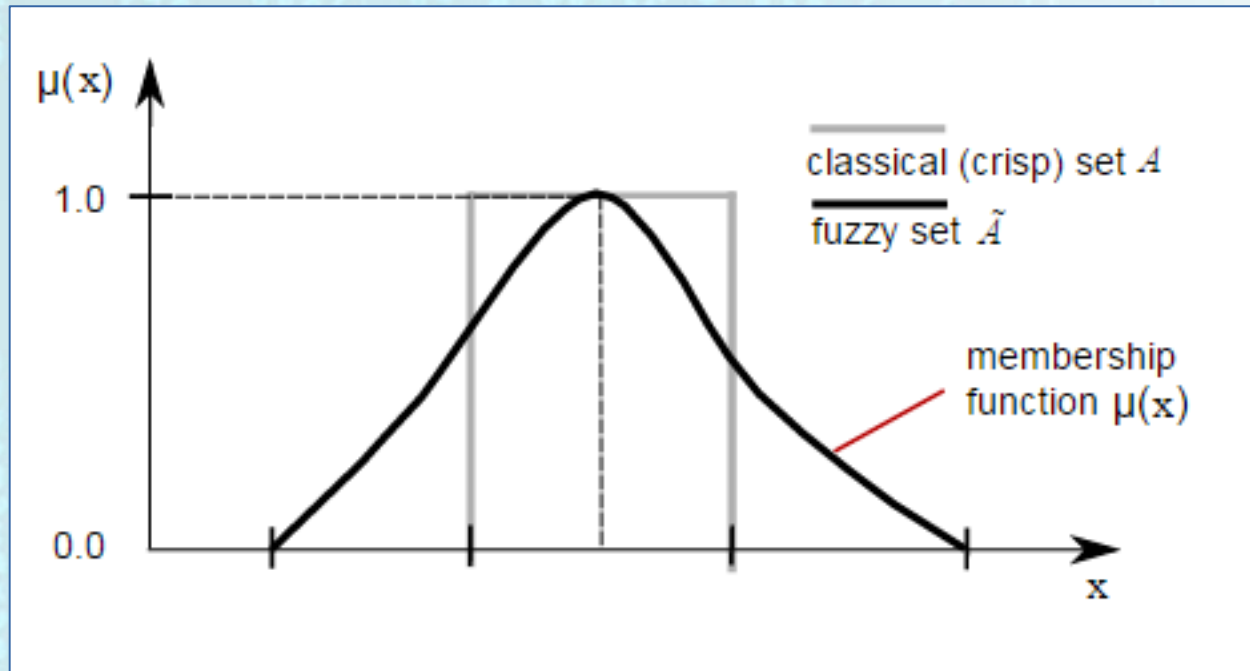


MEMBERSHIP FUNCTIONS^[14]

For any set X , a Membership function on X is any function from X to the real unit interval $[0,1]$.

Membership functions on X represent fuzzy subsets of X . The membership function which represents a fuzzy set A is usually denoted by μ_A . For an element x of X , the value $\mu_A(x)$ is called the membership degree of x in the fuzzy set A . The membership degree $\mu_A(x)$ quantifies the grade of membership of the element x to the fuzzy set A . The value 0 means that x is not a member of the fuzzy set; the value 1 means that x is fully a member of the fuzzy set. The values between 0 and 1 characterize fuzzy members, which belong to the fuzzy set only partially.

FUZZY LOGIC : AN INTRODUCTION



CONSTRUCTION OF MEMBERSHIP FUNCTIONS^{[13][15]}

- Intuition
- Rank Ordering
- Mathematical Modelling
- Adaptive Technique (Genetic Algorithm, Neural Networks etc).

INFORMATION SOURCES AS INPUTS

Input Vector, $\mathcal{S} = \{s^0, s^1, \dots, s^j, s^0\}$, where s^i takes values between 0 and S_{max}^i

For CAS System

$\mathcal{S} = \{DS, LS, VS, EFE, CE, PE, NE, ME, CO\}$

S_{max}^i for \mathcal{S} is chosen as the multiples of 5 for sake of simplicity

We can define more number of parameters for \mathcal{S} depending the requirement

MEMBERSHIP FUNCTION (DISCRETE VALUES)

Name	Rather Low	Low	Very Low	Very Very Low	Very Very High	Very High	High	Rather High
Acronym	$L_{-}^{S_{max}^i}$	$L_{--}^{S_{max}^i}$	$L_{---}^{S_{max}^i}$	$L_{----}^{S_{max}^i}$	$H_{++++}^{S_{max}^i}$	$H_{+++}^{S_{max}^i}$	$H_{++}^{S_{max}^i}$	$H_{+}^{S_{max}^i}$
$s^i \in [0; S_{max}^i/5]$	1	1	1	1	0	0	0	0
$s^i \in]S_{max}^i/5; 2 \cdot S_{max}^i/5]$	3/4	2/3	1/2	0	0	0	0	1/4
$s^i \in]2 \cdot S_{max}^i/5; 3 \cdot S_{max}^i/5]$	1/2	1/3	0	0	0	0	1/3	1/2
$s^i \in]3 \cdot S_{max}^i/5; 4 \cdot S_{max}^i/5]$	1/4	0	0	0	0	1/2	2/3	3/4
$s^i \in]4 \cdot S_{max}^i/5; S_{max}^i]$	0	0	0	0	1	1	1	1

The 8 membership functions (or fuzzy subsets) for an input s^i (S_{max}^i is the maximum value of this input)

ALGORITHM FOR CALCULATION OF ATTACK LEVEL

Method Chosen for inferring a decision from fuzzy rules and inputs (by *Mamdani*)

Algorithm 1 Algorithm for calculating the Misuse Level (ML)

Require: Inputs: Scalar values of the inputs (S)

Ensure: Output: Misuse Level (ML)

Require: Fuzzy Sets for inputs

Require: Fuzzy Sets for the outputs

Require: The set of rules

Fuzzify the values of the inputs

Compute the degree of truth of each rule of the rule set

Aggregate all the rules to obtain the membership function of the ML

Defuzzify this membership function to obtain the scalar value of the ML

Name	<i>LOW</i>	<i>HIGH</i>
$o \in [0; 0, 2]$	1	0
$o \in]0, 2; 0, 8]$	$-5/3 \cdot o + 2/3$	$5/3 \cdot o - 1/3$
$o \in]0, 8; 1]$	0	1

Membership functions for outputs

Membership values of outputs are continuous

RULES AND FUZZY OPERATIONS

- Expressed as “IF-THEN” rules
- “IF” term is *premise (or precondition)*
- “THEN” term is *conclusion*
- *Premises* are generally expressed as boolean operations on fuzzy sets

ZADEH OPERATORS ON FUZZY SETS^[11]

$$Z_NOT : \mu_{NOT(A_0)}(x) = 1 - \mu_{A_0}(x)$$

$$Z_AND : \mu_{AND(A_0, A_1)}(x, y) = \min(\mu_{A_0}(x), \mu_{A_1}(y))$$

$$Z_OR : \mu_{OR(A_0, A_1)}(x, y) = \max(\mu_{A_0}(x), \mu_{A_1}(y))$$

$$\mu_{AND(A_0, \dots, A_k)}(x_0, \dots, x_k) = \min(\mu_{A_0}(x_0), \dots, \mu_{A_k}(x_k))$$

$$= \min_{j=0}^k (\mu_{A_j}(x_j))$$

$$\mu_{OR(A_0, \dots, A_k)}(x_0, \dots, x_k) = \max(\mu_{A_0}(x_0), \dots, \mu_{A_k}(x_k))$$

$$= \max_{j=0}^k (\mu_{A_j}(x_j))$$

RULE SETS

R_0	IF the number of PIN code error is rather high OR the number of triggers of the voltage sensor is high THEN the misuse is high
	IF (PE is H_+ OR VS is H_{++}) THEN Misuse is HIGH
	IF $\mu_{OR(H_+,H_{++})}(PE,VS)$ THEN $\mu_{HIGH}(Misuse)$
R_1	IF the number of methods that have processed without error is high THEN the misuse is low
	IF (NE is H_{++}) THEN Misuse is LOW
	IF $\mu_{H_{++}}(NO_E)$ THEN $\mu_{LOW}(Misuse)$

Examples of rules set

- Membership Degree of a *premise* is a real number in [0,1]
- Depends on the values of S inputs
- Membership Degree of *premise* of rule i is denoted as $pre_i(S)$.

$$\begin{aligned}
 \text{For } R_0 \text{ and } R_1 \in R \quad \mu_{OR(H_+,H_{++})}(PE,VS) &= pre_0(S) \\
 \mu_{H_{++}}(NE) &= pre_1(S)
 \end{aligned}$$

RULE SETS

- We will distinguish rule sets for Attack Level / Misuse Level
- LOW value for misuse corresponds to “LOW-m” rules
- HIGH value for misuse corresponds to “HIGH-m” rules
- Consider set of rules , R with $R = \{R_0, R_1, \dots, R_p\}$
- We assume “LOW-m” rules consists of $\{R_0, R_1, \dots, R_{q-1}\}$
 “HIGH-m” rules consists of $\{R_q, R_{q+1}, \dots, R_p\}$

	Rule number	Rules
LOW-m	R_0	IF $pre_0(\mathbf{S})$ THEN Misuse is LOW

	R_{q-1}	IF $pre_{q-1}(\mathbf{S})$ THEN Misuse is LOW
HIGH-m	R_q	IF $pre_q(\mathbf{S})$ THEN Misuse is HIGH

	R_p	IF $pre_p(\mathbf{S})$ THEN Misuse is HIGH

Considered set of rules \mathcal{R}

DEGREE OF TRUTH OF THE RULES

COMPUTATION OF VALUES OF PREMISES

$$S^0_{max} = PE_{max} = 10 \quad S^1_{max} = VS_{max} = 10 \quad S^2_{max} = NE_{max} = 1000$$

Sensors	Case 1	Case 2	Case 3
$S^0 = PE$	3	5	9
$S^1 = VS$	6	2	8
$S^2 = NE$	300	900	700
$\mu_{H_+}(PE)$	1/4	1/2	1
$\mu_{H_{++}}(VS)$	1/3	0	2/3
$pre_0(S) = \mu_{OR(H_+, H_{++})}(PE, VS)$	1/3	1/2	1
$pre_1(S) = \mu_{H_{++}}(NE)$	0	1	2/3

Examples of degree of truth of the premises of R_0 and R_1

$$pre_i(S) \in P = \{ 0 ; 1/4 ; 1/3 ; 1/2 ; 2/3 ; 3/4 ; 1 \}$$

$\forall \mu_A(S = s^i)$ where μ_A is the membership function
for fuzzy set A of input S

DEGREE OF TRUTH OF THE RULES

MODIFICATION OF MEMBERSHIP FUNCTION OF THE CONCLUSION OF A RULE

- Degree of truth of the premise of a rule *modifies* the $\mu(\text{conclusion})$
- Modification \Rightarrow Comparing the $\mu_{A(k)}(y)$ with $pre_k(S)$
where $k \in \{ \text{Misuse/Attack Levels} \}$

$$\mu_{R_k}(y|S) = \min(pre_k(S), \mu_{A_k}(y))$$

Different rules are fired in parallel which might lead to inconsistency that is several rules could lead to different conclusions

UNIQUE SOLUTION

- Aggregation of different set of rules
- Defuzzification to compute a unique value for the decision

DEGREE OF TRUTH OF THE RULES

MODIFICATION OF MEMBERSHIP FUNCTION OF THE CONCLUSION OF A RULE (CASE 1 AND R_0)

Sensors	Truth of premise	Membership function of conclusion
$(s^0 = 3, s^1 = 6)$	$1/3$	

Name	LOW	HIGH
$o \in [0; 0, 2]$	1	0
$o \in]0, 2; 0, 8]$	$-5/3 \cdot o + 2/3$	$5/3 \cdot o - 1/3$
$o \in]0, 8; 1]$	0	1

Membership functions for outputs

$$pre_k(\mathbf{S}) = 1/3$$

$$\begin{aligned} \mu_{R_0}(0/\mathbf{S}) &= \min(1/3, \mu_H(0)) \\ &= \min(1/3, 0) = 0 \end{aligned}$$

$$\begin{aligned} \mu_{R_0}(0.2/\mathbf{S}) &= \min(1/3, \mu_H(0.2)) \\ &= \min(1/3, 0) = 0 \end{aligned}$$

$$\begin{aligned} \mu_{R_0}(0.33/\mathbf{S}) &= \min(1/3, \mu_H(0.33)) \\ &= \min(1/3, 2/9) = 2/9 \end{aligned}$$

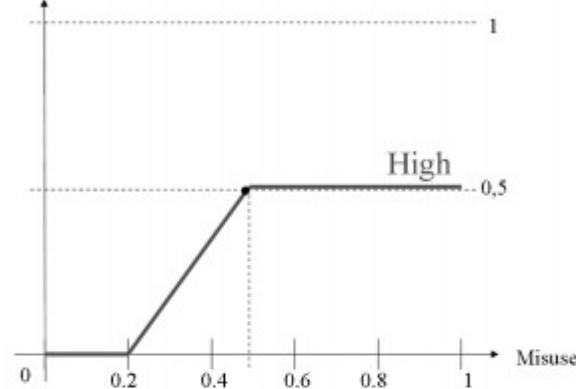
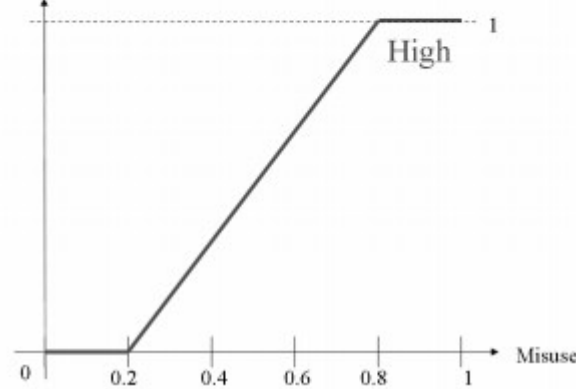
$$\begin{aligned} \mu_{R_0}(0.4/\mathbf{S}) &= \min(1/3, \mu_H(0.4)) \\ &= \min(1/3, 1/3) = 1/3 \end{aligned}$$

$$\begin{aligned} \mu_{R_0}(0.8/\mathbf{S}) &= \min(1/3, \mu_H(0.8)) \\ &= \min(1/3, 1) = 1/3 \end{aligned}$$

$$\begin{aligned} \mu_{R_0}(1/\mathbf{S}) &= \min(1/3, \mu_H(1)) \\ &= \min(1/3, 1) = 1/3 \end{aligned}$$

DEGREE OF TRUTH OF THE RULES

MODIFICATION OF MEMBERSHIP FUNCTION OF THE CONCLUSION OF A RULE (CASE 2,3 AND R_0)

$(s^0 = 5, s^1 = 2)$	$0,5$	<p style="text-align: center;">Membership functions</p> 
$(s^0 = 9, s^1 = 8)$	1	<p style="text-align: center;">Membership functions</p> 

DEGREE OF TRUTH OF THE RULES

MODIFICATION OF MEMBERSHIP FUNCTION OF

THE CONCLUSION OF A RULE (CASE 1,2,3 AND R₁)

Sensors	Truth of premise	Membership function of conclusion
$(s^2 = 300)$	0	

CASE 1: $pre_k(S) = 0$, the $\mu_{R_1}(y/S) = 0$
 for any values of $y \in (\text{Misuse/Attack Levels})$

$(s^2 = 900)$	1	
$(s^2 = 700)$	0,66	

DEGREE OF TRUTH OF THE RULES

AGGREGATION OF RULES

- Different rules are linked together with OR operator
- Combination consists of taking for all $y \in [0,1]$, the maximum value of the conclusion of the different rules

$$\mu_{\mathcal{R}}(y|\mathcal{S}) = \max_{k=0}^P(\mu_{R_k}(y|\mathcal{S}))$$

In the current scenario we combine both R_0 and R_1 to obtain the membership functions for each of the three cases.

DEGREE OF TRUTH OF THE RULES AGGREGATION OF RULES R_0 AND R_1 FOR CASE 1

Sensors	Truth of premise	Membership function of conclusion
$(s^0 = 3, s^1 = 6)$	$1/3$	

Sensors	Truth of premise	Membership function of conclusion
$(s^2 = 300)$	0	

For rule R_0

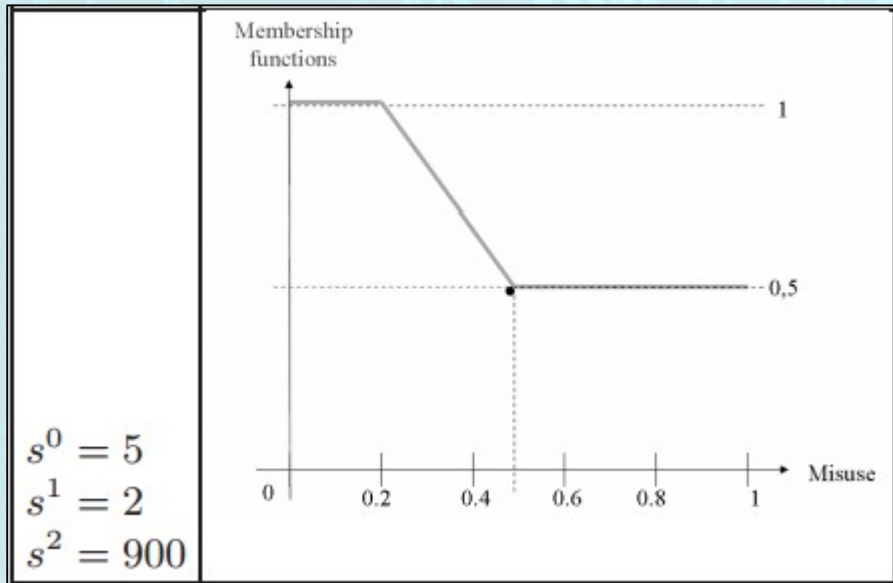
For rule R_1

Sensors	Membership function of conclusions
$s^0 = 3$ $s^1 = 6$ $s^2 = 300$	

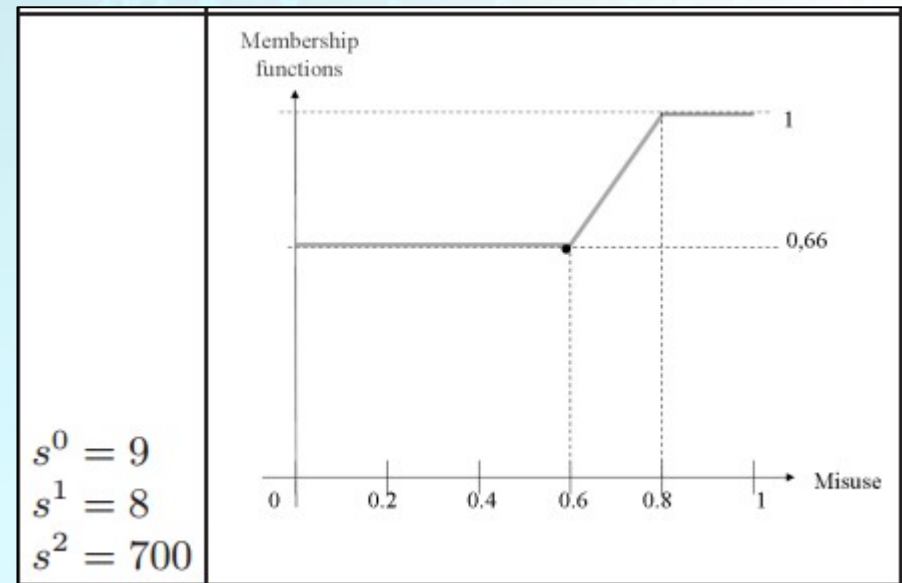
Combination of rules R_0 & R_1

DEGREE OF TRUTH OF THE RULES

AGGREGATION OF RULES R_0 AND R_1 FOR CASE 2, 3



(R_0, R_1) FOR CASE 2



(R_0, R_1) FOR CASE 3

OUTPUT MEMBERSHIP FUNCTION FOR ALL SET OF RULES

For Fuzzy Subset of “LOW-m”

Modification of Membership function

Aggregation of Rules

$$\begin{aligned} \mu_{LOW-m}(y|\mathcal{S}) &= \max(\min(pre_0(\mathcal{S}), \mu_{LOW}(y)), \\ &\dots, \min(pre_{q-1}(\mathcal{S}), \mu_{LOW}(y)) \\ &= \min(\max(pre_0(\mathcal{S}), \\ &\dots, pre_{q-1}(\mathcal{S})), \mu_{LOW}(y)) \\ &= \min(\max_{k=0}^{q-1}(pre_k(\mathcal{S})), \mu_{LOW}(y)) \end{aligned}$$

Similarly, for Fuzzy Subset of “HIGH-m”

$$\mu_{HIGH-m}(y|\mathcal{S}) = \min(\max_{k=q}^p(pre_k(\mathcal{S})), \mu_{HIGH}(y))$$

Let,

$$p_l = \max_{k=0}^{q-1}(pre_k(\mathcal{S})) \in P$$

$$p_h = \max_{k=q}^p(pre_k(\mathcal{S})) \in P$$

Then,

$$\mu_{LOW-m}(y)^{p_l} = \min(p_l, \mu_{LOW}(y))$$

$$\mu_{HIGH-m}(y)^{p_h} = \min(p_h, \mu_{HIGH}(y))$$

$$\text{OUTPUT } \mu_{\mathcal{R}}(y) \forall \mathcal{R}_i \in \mathcal{R} = \mu_{\mathcal{R}}(y)^{p_l, p_h} = \max(\mu_{LOW-m}(y)^{p_l}, \mu_{HIGH-m}(y)^{p_h})$$

DEFUZZIFICATION TECHNIQUES^[15]

Computation of Crisp Output from the output membership function

- Centroid Method
- Weighted Average Method
- Center of Sums
- Mean of Max (MofM)
- First of Max (FofM)
- Last of Max (LofM)

First of Max (FofM) =
Smallest element in of core (A)

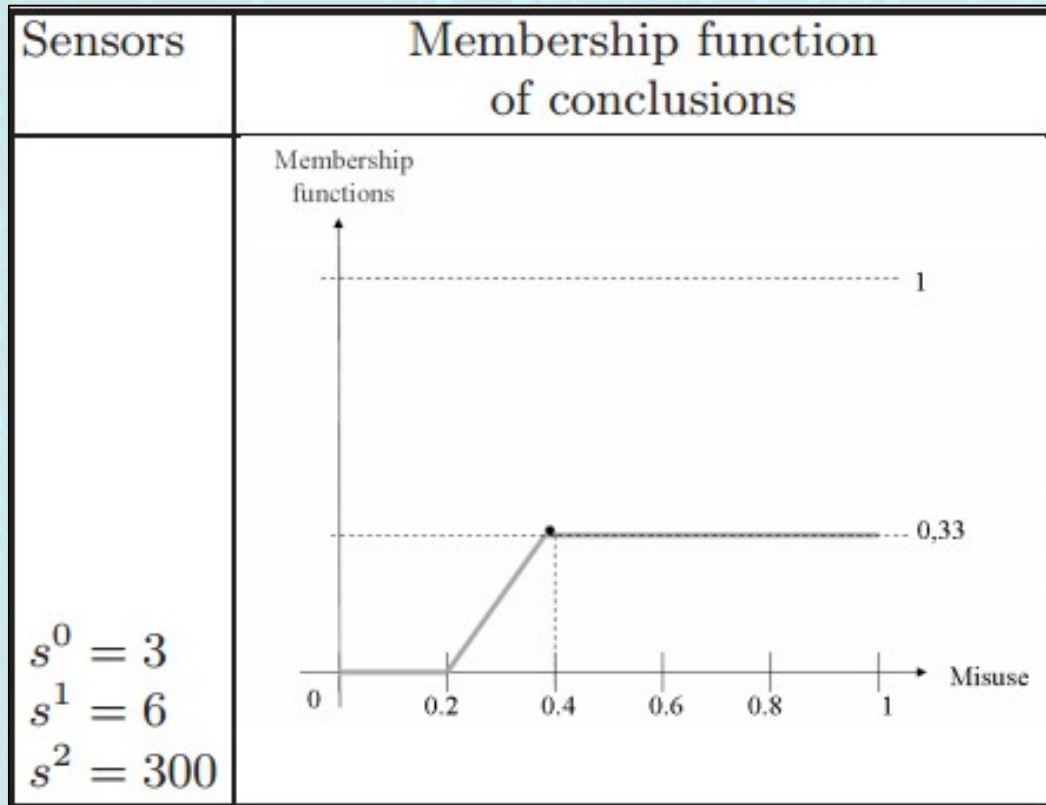
$$\mathbf{FofM = \min (core(A))}$$

The set of elements having the largest degree of membership in A is called the core of A^[16], i.e.,

$$\mathbf{core(A) = \left\{ x \in X \mid \mu_A(x) = \sup_{x \in X} \mu_A(x) \right\}}$$

Supremum or least upper bound of a set S of real numbers is denoted by *sup S* and is defined to be the smallest real number that is greater than or equal to every number in S.

DEFUZZIFICATION TECHNIQUE FOR CASE 1,2,3



CASE 1:

$$P_L = 0 ; P_H = 0.33$$

$$\text{FofM} = 0.4$$

Similarly,

For **CASE 2:**

$$P_L = 0.5 ; P_H = 1 ; \text{FofM} = 0$$

For **CASE 3:**

$$P_L = 0.66 ; P_H = 1 ; \text{FofM} = 0.8$$

Misuse Level	CASE 1	CASE 2	CASE 3
	0.4	0	0.8

CONFIRGURATION OF COUNTERMEASURES^[1]

- Total Number of RNG's for Random Power Generator = $R \in \{0;3;10\}$
- Total Redundancy Level Selected = $R_L \in \{1;2;3\}$
- Total Number of Useful Instuctions , $D \in \{0;4;8\}$
- Total Number of Dummy Instructions , $N \in \{2;3;4\}$
- Four Set of Countermeasures are defined (depends on the user)

Config.	Sensors	RL	RPG	IDI	Mute or Reset	Kill	FSCA	F DFA	Time	Energy
<i>Safe</i>	ON	×1	0	($D = 2; N = 0$)	No	No	1.0	1.0	1.0	1.0
<i>Unsafe</i>	ON	×2	3	($D = 3; N = 4$)	No	No	122.5	6270.7	4.0	5.2
<i>Critical</i>	ON	×3	10	($D = 4; N = 8$)	Yes	No	1346.7	1.0E+08	7.8	15.6
<i>Fatal</i>	-	-	-	-	-	Yes		-		

Countermeasure configurations

CONFIRGURATION OF COUNTERMEASURES

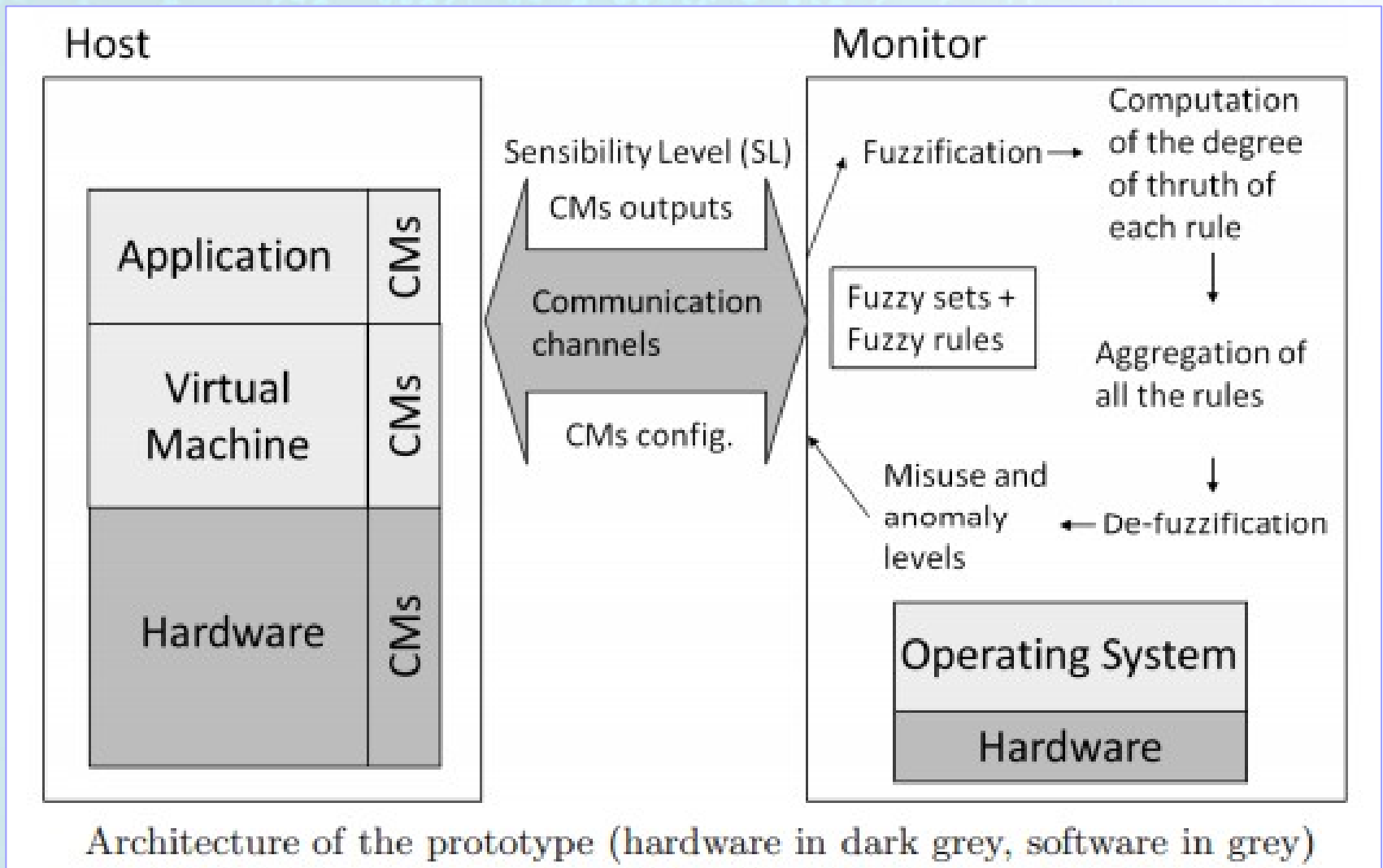
MISUSE/ATTACK LEVEL	COUNTERMEASURES
0.0 – 0.2	<i>Safe</i>
0.2 – 0.4	<i>Safe</i>
0.4 – 0.6	<i>Unsafe</i>
0.6 – 0.8	<i>Critical</i>
0.8 – 1.0	<i>Fatal</i>

In our given scenario we have the following configuration

	CASE 1	CASE 2	CASE 3
Misuse Level	0.4	0	0.8
Countermeasure	<i>Safe</i>	<i>Safe</i>	<i>Critical</i>

Similarly we can compute **Anomaly Level** from the above Fuzzy Technique and in combination with Misuse/Attack Level we can configure the countermeasures accordingly

HW/SW PROTOTYPE^[1]



HW/SW PROTOTYPE

- **Monitor** consists of : Software (Mini OS & Strategy of Security) and Hardware
- *Communication* between the host and the monitor is based on request / acknowledge protocol
- Host => Request => Waits for the Monitor to respond => Monitor acknowledge

Description of the protocol

- The Application indicates the variation in Sensitivity of data (**DS**)
- The Virtual Machine sends information about **DS** and Security Sensors via a *communication channel* (e.g. UART)
- The Host halts the current execution
- From the fuzzy sets and the fuzzy rules defined by the user, the monitor processes the inputs by fuzzy reasoning (as described in previous slides)
- The outputs of the reasoning (e.g. Misuse/Attack Level) selects the configuration of countermeasures
- The monitor asks the host system to incorporate the countermeasure and reconfigure the parameters.
- The monitor waits till the configurations are done and ready
- The monitor then again waits for the next information set and the process continues

SIMULATION SCENARIOS^[1]

Experiment : Impact of Laser Attack on Strategy of Security

OBSERVATIONS

PART I:

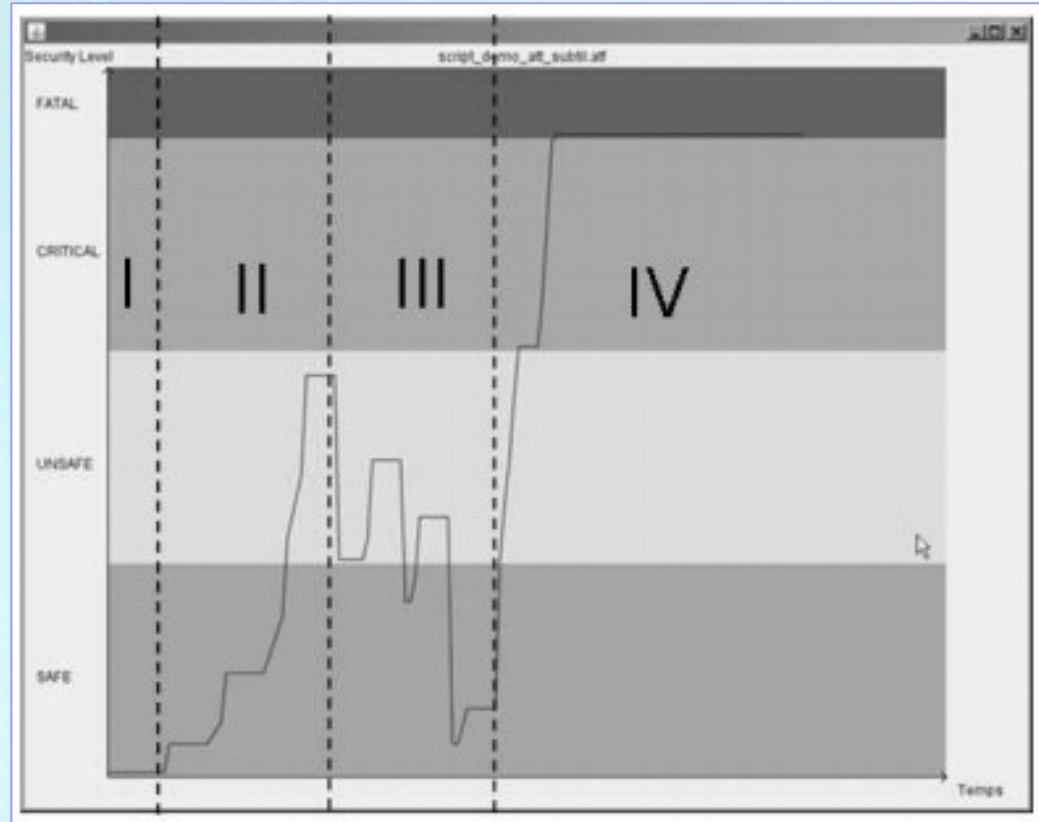
Initially No errors and the security level remains low.

PART II:

The light sensors are triggered as the adversary injects faults in the middle of long correct sequences. The security level increases rapidly

PART III:

The adversary somehow able to analyze the security level (e.g increase in the value of R for RPG); stops injecting faults. The security level tends to decrease



PART IV: After sometime, when the adversary resumes the attack, the security level increases abruptly leading to deletion of sensitive data

LIMITATIONS & CHALLENGES

- Area consumption during hardware implementation of security of strategy. Can be dealt with design optimization.
- Rate of Communication channels between the host and monitor, rate of change of countermeasure strategy or the time, the host requires to reconfigure, affects the performance of the strategy.
- Design of the best strategy for a given application i.e. identifying the correct set of inputs and outputs, design of efficient membership functions and design the level of countermeasures to be implemented catering all the possible scenario in “real-time” environment.
- Strategy should not only aim at the trade-off between the performance and security according to application's constraints but also reduce the number of false triggers.
- Complexity for Testing and Debugging of the whole prototype needs to handled at unit as well as system level

CONCLUSION

- “System Level” Management of the security dedicated to the improvement of the availability and the performance with security.
- Impact in terms of Security and Performances of different well-known countermeasures are quantified.
- Modification of few parameters of countermeasures can lead to states with distinct performance and security levels.
- Strategy of Security implementation to minimize rate of anomalies considered to be attacks and rate of attacks considered to be normal.
- Dynamic Management for Strategy of Security using Fuzzy Approach
- HW/SW architecture which essentially divides the design into two parts in a way that no sensitive data is requested while applying strategy

REFERENCES

- [1] Smart Security Management in Secure Devices by Bruno Robisson, Michel Agoyan, Patrick Soquet, Sébastien Le Henaff, Franck Wajsbürt, Pirouz Bazargan-Sabet, Guillaume Phan , 2015/670,Cryptology ePrint Archive
- [2] Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 1666, pp. 388–397. Springer (1999)
- [3] Stefan Mangard, E.O., Popp, T.: Power Analysis Attacks Revealing the Secrets of Smart Cards. Springer Verlag (2007)
- [4] Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: Kaliski Jr., B. (ed.) Advances in Cryptology - CRYPTO '97. Lecture Notes in Computer Science, vol. 1294, pp. 513–525. Springer (1997)
- [5] Robisson, B., Manet, P.: Differential behavioral analysis. In: Paillier, P., Verbauwhede, I. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2007, Lecture Notes in Computer Science, vol. 4727, pp. 413–426. Springer Berlin /Heidelberg (2007),
http://dx.doi.org/10.1007/978-3-540-74735-2_28, [10.1007/978-3-540-74735-2_28](https://doi.org/10.1007/978-3-540-74735-2_28)
- [6] Smart Cards: A Case Study by Jorge Ferrari,Robert Mackinnon,Susan Poh and Lakshman Yatawara
IBM :<http://www.redbooks.ibm.com/redbooks/pdfs/sg245239.pdf>

REFERENCES

[7]Java card technology website, <http://java.sun.com/javacard>

[8]Global platform specifications website, <http://www.globalplatform.org>

[9]Ambrose, J.A., Ragel, R.G., Parameswaran, S.: RIJID: Random Code Injection to Mask Power Analysis based Side Channel Attacks. In: Proc. Design Automation Conference – DAC, ACM. pp. 489–492 (2007)

[10] Clavier, C., Coron, J.S., Dabbous, N.: Differential power analysis in the presence of hardware countermeasures. In: Proceedings of the Second International Workshop on Cryptographic Hardware and Embedded Systems. pp. 252–263. CHES '00, Springer-Verlag, London, UK (2000)

[11] Zadeh, L.: Fuzzy logic. Computer 21(4), 83 –93 (Apr 1988)

[12] Fault Analysis in Cryptography,Editors: Marc Joye, Michael Tunstall
ISBN: 978-3-642-29655-0 (Print) 978-3-642-29656-7

[13] Introduction to Fuzzy Logic using MATLAB By S.N. Sivanandam, S. Sumathi, S. N. Deepa

[14] [https://en.wikipedia.org/wiki/Membership_function_\(mathematics\)](https://en.wikipedia.org/wiki/Membership_function_(mathematics))

[15] <http://www.csee.wvu.edu/classes/cpe521/presentations>

[16] Parameterized defuzzification with maximum entropy weighting function—Another view of the weighting function expectation method by Xinwang Liu;doi:10.1016/j.mcm.2006.04.014



THANK YOU