Security Measurement in Secure Smart Card

Rakesh Kumar Jangid[1], Uma Choudhary[2], Swapnesh Taterh[3]
Research Scholar RTU KOTA, Research Scholar Jaggannath University, Rearch Scholar Pacific University Udaipur Jaipur,Rajasthan (India)
[1]rakesh283343@gmail.com, [2]umasweetuma@rediffmail.com, [3]Swapnesh_t@yahoo.com

***Abstract* -To deal with software security issues in the early stages of system development, this paper presents a Secure System development Life Cycle Approach. In this approach, we analyze the security requirement using security and functional requirement .Design secure system and architecture of smart card using use case, Attack Tree, Threat Modeling and risk assessment process. Analyzing static analysis and dynamic analysis of smart card microcontroller and finally applies the various securities testing approach. In this paper we estimate the security improvement due to secure SDLC of Smart card using the Semi-Marchov Chain and Steady state matrix for different attack levels. To analyze the security of any driver of smart card we use Security State Diagram. Security improvement is computed due to security levels difference between integrated and series system security.**

***Keywords* - CIAAN, GTM, SC, SDLC, p_v, p_c, p_I, p_A, p_{Aut}, P_{Nrp}, P_{CIAAN}. Use Case Diagram, Attack Tree, Threat Modeling, Vulnerabilities, Risk Mitigation, Security State Diagram, semi-Markov chain, Generic Transition Matrix, Steady State Security.**

I . INTRODUCTION

A smart card is a device that includes an embedded integrated circuit that can be either a secure microcontroller or equivalent intelligence with internal memory or a memory chip alone. The card connects to a reader with direct physical contact or with a remote contactless radio frequency interface. With an embedded microcontroller, smart cards have the unique ability to store large amounts of data, carry out their own on-card functions (e.g., encryption and mutual authentication) and interact intelligently with a smart card reader. Different kinds of smart cards are there- Memory Cards, Microprocessor Cards, and Contactless Cards.

II . REQUIREMENT ELICITATION

A. *Assets Identification*

1). Hardware components: Servers, Memory, Communication channels and lines, Hardware tokens.

2). Software components: Operating systems, Database management systems, and Communication and security application programs.

3). Data: Data, including databases containing customer - related information.

4). Personnel: Professional personnel, clerical staff, administrative personnel, and computer staff.

B. *Security functional requirements*

Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform.

1). Security Testing: User authentication before any action, User Identification before any action, User Attribute Definition, Stored Data Integrity Monitoring.

2). *Security Management :* Management of security functions behavior, Management of security attributes, Security roles, Static Attribute Initialization, Complete Access Control, Security Attribute Based Access Control, Subset Information Flow Control, Simple Security Attributes, Potential Violation Analysis, Unobservability, Notification of Physical Attack, Resistance to Physical Attack.

III . SECURE DESIGN

Secure design of smart card contains following properties-

A. *Use Case Diagram*

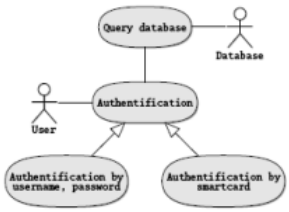


Fig.1 Use Case Diagram of Smart Card

B. *Attack Tree*

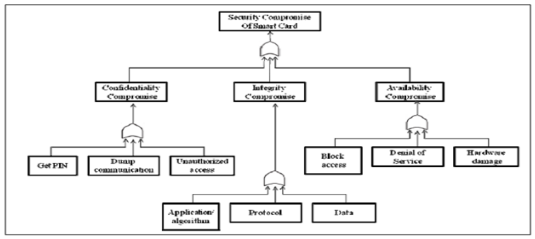


Fig.2 Attack Tree of Smart Card

As shown in the figure 3, confidentiality of the smart card is compromised by stealing its PIN, sniffing and unauthorized access. Integrity is generally violated when the attacker can exploit a badly written protocol or an unsecure application or use of inefficient cryptographic technique on the data. Similarly availability is compromised by blocking PIN access, denial of service and hardware damage.

C. *Threat Modeling*

A threat is the potential for a particular threat source to practice vulnerability; threats are the possible means by which a security policy may be breached. A threat source can be any person, thing, event, or idea that poses danger to an asset within a system in terms of confidentiality, integrity, availability, or legitimate use. Possible threats on the smart card system can be –Unauthorized system access, Hacking and System intrusion, Information leakage or theft, Integrity violation, Availability violation like Distributed Denial of Service, Illegitimate use, System penetration and tampering.

D. *Attacks on Smart Card*

All attacks that can be in smart card are shown in table.

E. *Vulnerability Identification*

1). *Technical Vulnerabilities:* TCP/IP protocol stack, Lack of database backup, Weak user authentication and authorization methods, Old encryption methods, Firewall allows inbound telnet, and guest ID is enabled on the organization's server.

2). *Non-technical Vulnerabilities:* Negligence in removing the terminated employees' system identifiers, Carelessness of monitoring employees' behavior.

F. *Risk Determinations*

Risk is the measure of the cost of vulnerability taking into account the probability of a successful attack. The risk management program consists of three main processes, the processes are: Risk assessment, Risk Matrix, Risk mitigation.

1). *Risk assessment:* It determines the risk level and examines the risk probability and impact associated with each type of smart card. The level of risk is measured by a risk value; this value could be described as high, medium, or low.

2). *Risk Matrix:* Developing a risk matrix that shows the risk tolerability depending on the probability of attack taking place and the consequence of this attack on each type of smart card. In order to construct the risk matrix, the likelihood of occurrence and the consequence of any attack to the smart card must be considered. The probabilities of attacks (Pa) on each type of smart card along with the consequence of attacks (Ca) are going to be assigned. Table (1) shows the weights and their descriptions.

TABLE 1 Probability of attack and Consequence of attack

Weights	Probability of Attack (Pa)	Consequences of Attack(Ca)
0 to .2	Improbable	Negligible
.3 to .5	Occasional	Marginal
.6 to .8	Probable	Critical
.9 and above	Frequent	Catastrophic

Weights are assigned to the smart card types to determine the risk level. The main purpose of developing the risk matrix is to show the risk tolerability level that each smart card type has.

TABLE 2 Risk Matrix

Consequences Probability	Negligible (0≤Ca≤.2)	Marginal (.3≤Ca≤.5)	Critical (.6≤Ca≤.8)	Catastrophic (.9≤a)
Frequent (.9≤Pa)			Identification Card	
Probable (.6≤Pa≤.8)		Identification Card	Banking Card	
Occasional (.3≤Pa≤.5)		Health Card		
Improbable (0≤Pa≤.2)	Loyalty Card	Prepaid Card		

3). *Risk Mitigation:* It is extremely important to mitigate or even trying to eliminate the risks. The main idea behind this part of the process is to come up with controls to eliminate the risk or reduce the level of risk to an acceptable level.

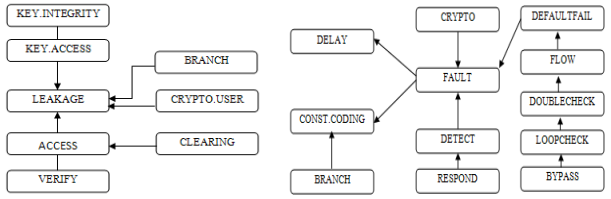
TABLE 3 Security Methods for each type of Smart Card

Banking Card	Identification Card	Health Card	Loyalty Card	Prepaid Card
PIN and Biometrics (Fingerprint or Signature)	PIN and Biometrics (Fingerprint or Signature)	PIN and Biometrics (Fingerprint or Signature)	PIN	PIN
Cryptographic Key Management (PKI)	Cryptographic Key Management (PKI)	Cryptographic Key Management (PKI)	Symmetric or Asymmetric Encryption And Digital Signature	Asymmetric Encryption And Digital Signature

G. *Secure Design Pattern*

1). *Patterns to defend against data leakage*

The secure design patterns are used to prevent application program code from leaking sensitive information when this information is processed. Application developers can use these patterns to protect confidential data like keys and passwords.



Patterns to defend against data leakage *Patterns to defend against fault injection*

Fig.3 Secure Design Pattern for Smart Card

A. *Patterns to defend against fault injection*

The patterns in this section assist in preventing fault injection in application program code and in responding to it. These patterns protect critical data or program flow.

IV . SECURE CODING

During the coding of embedded software and operating system following coding practices can be helpful.

A. Strong typing

Strong typing in programming languages imposes a discipline on the use of data and code: a reference cannot be forged by casting from an integer; one cannot jump in the middle of the code of a function. This discipline is good for program reliability in general. But it can also be exploited to ensure that a software isolation layer cannot be bypassed. There are many other ways by which type-unsafe code can cause the virtual machine to malfunction and break isolation: Pointer forging, Illegal cast, Out-of-bounds access, and Stack smashing, Context switch prevention, explicit deallocation

B. Static analyses

To prevent such breaches of confidentiality, attach information levels to every variable, Abstract the flow of control as a method call graph, Express control properties of interest in temporal logic, check these properties on the abstraction using model checking.

C. Dynamic analyses

Protection: monitoring the passivation layer, Protection voltage monitoring, Protection frequency is monitoring, Protection bus scrambling, and Protection irreversible switching from the test mode to the user mode.

V . SECURE TESTING

Protecting smartcard chips and other embedded microchips against unauthorized access is one of the main security challenges facing the smartcard industry. The goal of the development was to create a video based, high power microscope instrument capable of delivering and imaging an extremely small, highly focused area of 2 wavelengths of laser energy to localized areas of a secure microcontroller, and then to give the possibility to reposition the chip with very high accuracy.

I . SECURITY MEASUREMENT

An attacker's behavior can be unpredictable and random in nature and may take more than one form. The semi-Markov chain process is considered to be an appropriate modeling tool to illustrate the behavior of attackers. The semi-Markov chains have a special property in that the probability of any event moving to a future state depends only on the present state. If the present state denotes Xt = i, then the future state will be Xt+1 = i + 1, and the past state is Xt-1 = i - 1. The n-step transition probabilities pij(n) is the conditional probability that the system will be in state j = i + 1 (future state) after exactly n steps, given that it starts in state i at any time t, is pij = P{Xt+n = j| Xt = i}, and for n transition steps is pij(n) = P{Xt+n = j| Xt = i} . Because the pij(n) are conditional probabilities, they must be positive, and therefore must make a transition into some state where the conditions of pij(n) ≥ 0 satisfy all the values for i and j; and $\sum_{j=0}^{N} p_{ij}^{(n)} = 1,$

Where n = 0, 1, 2 …, N.

A. States

0: Normal	5: Availability attacked
1: Vulnerable	6: Authentication attacked
2: Attacked	7: Non-Repudiation
3: Confidentiality attacked	8: Failure
4: Integrity attacked	

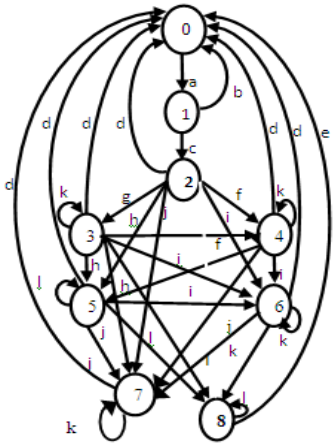


Fig.4 Security State Diagram of a driver under attack

B. Transitions

a: Vulnerability found	g: Attack on confidentiality
b: Vulnerability eliminated	h: Attack on availability
c: Vulnerability exploited	i: Attack on authentication
d: Attack & recovery initiated	j:Attack on Non-Repudiation
e: System restarted	k: Attack & recovery failed
f: Attack on integrity	l: System failed

From all states, the system can return back to state (0), the normal state, with different levels of probability and degrees of loss. P in the above is a matrix formulation of the relationship between the states and their probability.

$$P = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \bar{p}_1 & 0 & p_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ \bar{p}_2 & 0 & 0 & p_{23} & p_{24} & p_{25} & p_{26} & p_{27} & 0 \\ \bar{p}_3 & 0 & 0 & p_{33} & p_{34} & p_{35} & p_{36} & p_{37} & p_{38} \\ \bar{p}_4 & 0 & 0 & 0 & p_{44} & p_{45} & p_{46} & p_{47} & p_{48} \\ \bar{p}_5 & 0 & 0 & 0 & 0 & 0 & 0 & p_{57} & p_{58} \\ \bar{p}_6 & 0 & 0 & 0 & 0 & 0 & p_{66} & p_{67} & p_{68} \\ \bar{p}_7 & 0 & 0 & 0 & 0 & 0 & 0 & p_{77} & p_{78} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Where

$\bar{p}_{1+} p_{12=1}$

$\bar{p}_{2+} p_{23+} p_{24+} p_{25+} p_{26+} p_{27=1}$

$\bar{p}_{3+} p_{33+} p_{34+} p_{35+} p_{36+} p_{37+} p_{38=1}$

$\bar{p}_{4+} p_{44+} p_{45+} p_{46+} p_{47+} p_{48=1}$

$\bar{p}_{5+} p_{57+} p_{58=1}$

$\bar{p}_{6+} p_{66+} p_{67+} p_{68=1}$

$\bar{p}_{7+} p_{77+} p_{78=1}$

TABLE 4 State Transitions	
0 [a] 1	Moves to vulnerable state.
From State 1 (Vulnerable)	
1 [b] 0	Vulnerability eliminated
1 [c] 2	Vulnerability exploited
From State 2 (Attacked)	
2 [d] 0	Attack detected & recovery initiated
2 [g] 3	attack on confidentiality
2 [f] 4	attack on integrity
2 [h] 5	attack on availability
2 [i] 6	attack on authentication
2 [j] 7	attack on Non-Repudiation
From State 3 (Confidentiality attacked)	
3 **d** 0	Attack detected & recovery initiated
3 [k] 3	Attack detected and recovery failed
3 [f] 4	attack on integrity
3 [h] 5	attack on availability
3 [i] 6	attack on authentication
3 [j] 7	Attack on Non-Repudiation
3 [l] 8	System failed
From State 4 (Integrity attacked)	
4 [d] 0	Attack detected & recovery initiated
4 [k] 4	Attack detected and recovery failed
4 [h] 5	Attack on availability
4 [i] 6	attack on authentication
4 [j] 7	attack on Non-Repudiation
4 [l] 8	System failed
From State 5 (Availability attacked)	
5 [d] 0	Attack detected & recovery initiated
5 [i] 6	attack on authentication
5 [j] 7	attack on Non-Repudiation
5 [l] 8	System failed
From State 6 (Authentication attacked)	
6 [d] 0	Attack detected & recovery initiated
6 [k] 6	Attack detected and recovery failed
6 [j] 7	attack on Non-Repudiation
From State 7 (Non-Repudiation)	
7 [d] 0	Attack detected & recovery initiated
7 [k] 7	Attack detected and recovery failed
8 [e] 0	System restarted

C. *Security Analysis of Smart Card*

A smart card consists of four drivers as Hardware, Software, Data Transmission and people.

1) *CIAAN probabilities for Smart Card Drivers*: Drivers contains following values for Confidentiality, Integrity, Availability and Accountability. Driver 1, the Hardware, is concerned more with confidentiality and integrity, so high weights are given to both (pc = pI= 0.25). Driver 2, the software, is also concerned more with confidentiality and authentication, therefore a very high value is assigned for (pc = pAut = 0.20). Driver 3, the Data Transmission, is concerned more with Confidentiality, Integrity and Authentication therefore a high value of (pc = pI = pAut = 0.20) is assigned. Finally, driver 4, the people has a high confidentiality and Integrity, Availability and Authentication therefore high values (pc = pI = pA = pAut = 0.15) are assigned.

TABLE 5 CIAAN probabilities for Smart Card Drivers

Smart Card Driver	p_v	p_c	p_I	p_A	p_{Aut}	P_{Nrp}	$P_{\overline{CIAAN}}$
Hardware	.10	.25	.25	.10	.10	.05	.15
Software	.15	.20	.10	.10	.20	.05	.20
Data Transmission	.10	.20	.20	.10	.20	.10	.10
People	.15	.15	.15	.15	.15	.10	.15

2) *Generic transition matrix (GTM)*

The generic transition matrix GTM for each driver (***i***) created by substituting the parameters of $p_v,p_c,p_I,p_A,p_{Aut},P_{Nrp},P_{CIAAN}$ from Table 3. Table 4 presents a generic transition matrix (GTM) for a driver ***i*** created by substituting the parameters. (GTM) for a driver i created by substituting the parameters.

TABLE 6 Generic Transition Matrix for driver i

From \ To	N (0)	V (1)	Att (2)	C (3)	I (4)	A (5)	Aut (6)	Nrp (7)	F (8)
(0)	0	1	0	0	0	0	0	0	0
(1)	$\overline{p_1}$	0	p_{12}	0	0	0	0	0	0
(2)	$\overline{p_2}$	0	0	p_{23}	p_{24}	p_{25}	p_{26}	p_{27}	0
(3)	$\overline{p_3}$	0	0	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}	p_{38}
(4)	$\overline{p_4}$	0	0	0	p_{44}	p_{45}	p_{46}	p_{47}	p_{48}
(5)	$\overline{p_5}$	0	0	0	0	0	0	p_{57}	p_{58}
(6)	$\overline{p_6}$	0	0	0	0	0	p_{66}	p_{67}	p_{68}
(7)	$\overline{p_7}$	0	0	0	0	0	0	p_{77}	p_{78}
(8)	1	0	0	0	0	0	0	0	0

Table 7 presents an initial transition matrix for driver 1 (P1).it is generated by substituting the values of pv,pc,pI,pA,pAut,PNrp .

TABLE 7 GTM for driver 1 Pa1 at attack .10

From \To	N (0)	V (1)	Att (2)	C (3)	I (4)	A (5)	Aut (6)	Nrp (7)	F (8)
(0)	.90	.10	0	0	0	0	0	0	0
(1)	.90	0	.10	0	0	0	0	0	0
(2)	.25	0	0	.25	.25	.10	.10	.05	0
(3)	.15	0	0	.25	.25	.10	.10	.05	.10
(4)	.40	0	0	0	.25	.10	.10	.05	.10
(5)	.85	0	0	0	0	0	0	.05	.10
(6)	.75	0	0	0	0	0	.10	.05	.10
(7)	.85	0	0	0	0	0	0	.05	.10
(8)	1	0	0	0	0	0	0	0	0

Similarly generic transition matrix for driver 2, 3, 4 can be generated for different attack levels (0.10 to 0.90).

3) *Steady state security*

The steady states of a Smart Card can be achieved by multiplying the matrices of all drivers. The steady-state probability of a Smart Card, pS and CIAAN, could be calculated using the following relationships: pS = p0 + p1, pc = 1 - p3, pI = 1 - p4, pA = 1 - (p5 + p8), pAut = 1 – p6, PNrp = 1-p7. The steady-state security for driver 1, given for

eight attack levels, is summarized in Table 6. Hence, each row in this table represents a steady-state for corresponding attack level. Where pS = p0 + p1. Similarly steady-state security for driver 2, 3 and 4 are developed.

TABLE 8 Steady State Security for Driver 1

Attacker Level	Normal Π_0	V Π_1	Att Π_2	C Π_3	I Π_4	A Π_5	Aut Π_6	Nrp Π_7	F Π_8	Security Π_s
0.10	.8600	.0860	.0086	.0028	.0038	.0015	.0017	.0009	.0010	.9460
0.30	.8580	.0858	.0257	.0085	.0114	.0045	.0050	.0032	.0035	.9438
0.50	.8200	.0820	.0410	.0136	.0182	.0072	.0081	.0046	.0051	.9020
0.70	.7950	.0795	.0556	.0185	.0247	.0099	.0110	.0063	.0070	.8745
0.90	.7700	.0770	.0693	.0231	.0308	.0123	.0137	.0078	.0087	.8470

4) Integrated Steady-state Security

Table 9 represents system security when all drivers are sharing both the functional and the security information as an integrated security system.

TABLE 9 Integrated Steady-state Security for Smart Card

Attacker Level	Normal Π_0	V Π_1	Att Π_2	C Π_3	I Π_4	A Π_5	Aut Π_6	Nrp Π_7	F Π_8	Security Π_s
0.10	0.8800	0.0880	0.0044	0.0019	0.0016	0.0008	0.0008	.0025	0.0004	0.9621
0.30	0.8732	0.0873	0.0174	0.0074	0.0062	0.0031	0.0034	.0073	0.0016	0.9269
0.50	0.8481	0.0848	0.0296	0.0127	0.0106	0.0053	0.0058	.0116	0.0028	0.8887
0.70	0.8240	0.0824	0.0412	0.0176	0.0147	0.0073	0.0081	.0156	0.0039	0.8587
0.90	0.8012	0.0801	0.0520	0.0223	0.0186	0.0093	0.0103	.0190	0.0050	0.8249

D. Discussion of Results

To analyze the relationship between the four drivers of the Smart Card with system wide security there are two cases:

CASE 1: Drivers share information without sharing security and vulnerability information. In this case, system-wide security will be very low. The total security is a multiplication of all individual driver security values; Ps(sys) = pS1 pS2 pS3 pS4 (1).

CASE 2: Drivers share functional information as well as security information. In this case, the level of vulnerability will be reduced, hence increasing the security level. The integrated system is obtained mathematically by multiplying each master transition matrix for all drivers to obtain the transition matrix for the system, Psw as Psw = Pa1. Pa2. Pa3. Pa4.

1) Integrated Smart Card versus Individual Driver

Figure 4, the curve for system wide (SW) security shows improvements for all levels of attack (10% - 90%).These curves show much lower security for all drivers except for Hardware. On the other hand, when each driver represents an individual function in which security information is not shared, each driver will be more vulnerable to attack.

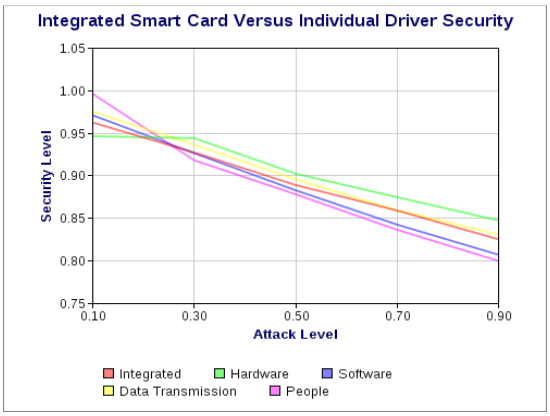


Fig.5 Integrated Smart Card versus Individual Driver

2) Comparison of System Security

Table 8 summarizes the results obtained from integrated and series system security of a Smart Card. We can clearly observe that security has improved by 37% (from 0.4537 to 0.8249 at the high attack level 90%).

TABLE 10 Comparison of System Security

Attacker Level	Type of System Security	
	Integrated	Series
0.10	0.9621	0.8914
0.30	0.9269	0.7509
0.50	0.8887	0.6256
0.70	0.8587	0.5290
0.90	0.8249	0.4537

State-wide system security for Smart Card has less vulnerability, which leads to better security due to the sharing of information about attackers. An individual driver is more vulnerable than the integrated system in a Smart Card. Figure 5 compares two curves; one represents integrated security for a Smart Card, and the other represents Smart Card drivers working together but without sharing security information are much more vulnerable.

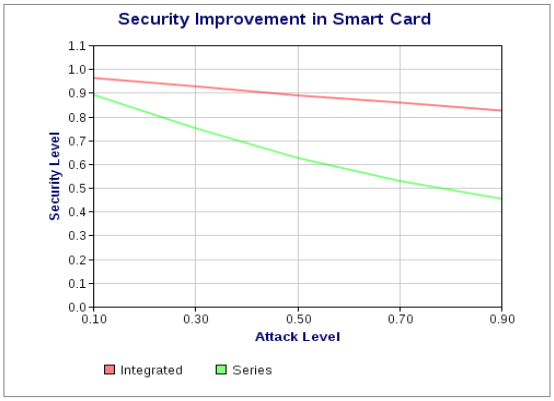


Fig.6 Security Improvement in Smart Card

TABLE 11 Attacks in Smart Card

Attack Class	Attack Type	Attack Class	Attack Type
Physical Attacks	1. Chemical Solvents, Etching and Staining Materials	**Logical Attacks**	1. Hidden Commands
	2. Static Analyses of the Smart Card Microcontroller		2. Buffer Overflow
	3. Dynamic Analyses of the Smart Card Microcontroller		3. File Access
	4. Probe Stations		4. Malicious Applets
	5. Focused Ion beam (FIB)		5. Communication Protocol
			6. Crypto-Protocol, Design ,Implementation
Side Channel Attacks	1. Timing Attack	**Development**	1. Development of the Smart Card Microcontroller
	2. Fault Attack		2. Development of Smart Card Operating System
	3. Power Analysis Attack	**Manufacturing Process**	1. Authentication in the Manufacturing Steps
	4. EM Attack	**Other Attacks**	1. Eavesdropping
	5. Acoustic Attack		2. Interruption of operations
	6. Visible Light Attack		3. Denial of service
	7. Error Message Attack		4. Covert transactions
	8. Cache-based Attack		5. Communication links and dual modes
	9. Frequency-based Attack		6. Data Remanence
	10. Scan-based Attack		7. Pin guessing
	11. Combination of Side Channel Attacks		8. Reverse engineering of the chipset
	12. Combination of SCA and Mathematical Attacks		

II . References

[1]. Security Measurement White Paper V3.0 13 January 2006 Prepared on behalf of the PSM Safety & Security TWG

[2]. www.4shared.com/office/.../John_Wiley__Smart_Card_.html

[3]. A Smart Card Alliance Contactless and Mobile Payments Council White Paper Publication Number: CPMC-08002 www.smartcardalliance.org.

[4]. Dependability and Security Models Kishor S. Trivedi, Dong Seong Kim, Arpan Roy. Department of Electrical and Computer Engineering. Duke University. Durham, NC, USA.www.sis.pitt.edu/~dtipper/3350/paper6.pdf.

[5]. Analysis of Information Security in Supply Chain Management Systems http://www.idea-group.com.

[6]. Build Security in Home https://buildsecurityin.us-cert.gov/.

[7]. Analysis of Information Security in Supply Chain Management Systems http://www.idea-group.com.

[8]. Smartcard IC Platform Protection Profile Version 1.0 2001.www.commoncriteriaportal.org/files/ppfiles/ssvgpp01.pdf.

[9]. (ISC)² - IT Certification and Security Experts www.isc2.org/.

[10]. The Open Web Application Security Project (OWASP) https://www.owasp.org/

[11]. Security Measurement White Paper V3.0 13 January 2006 Prepared on behalf of the PSM Safety & Security TWG

[12]. www.4shared.com/office/.../John_Wiley__Smart_Card_.html

[13]. A Smart Card Alliance Contactless and Mobile Payments Council White Paper Publication Number: CPMC-08002 www.smartcardalliance.org.