ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2017

A New Design Approach for Mobile Devices Security against IMEI Tampering and Cloning

Naveen Jakhar

Indian Telecommunication Service

Department of Telecommunications, Government of India, Ministry of Communications, India

Abstract: The advancements in VLSI design technologies, economy of scale, mass production, reduced prices of mobile devices and aggressive competition in the market have made mobile phones affordable to common man and it helps him to stay connected with his near and dear ones at anytime and anywhere. The IMEI for GSM/UMTS/LTE phones and ESN for CDMA phones is the unique identity of any mobile device. After November 2008, ESN has been replaced by MEID in CDMA mobile devices. But along with the genuine mobile devices, rogue and sub-standard mobile devices are being sold in the markets which are either having illegal IMEI or cloned IMEI. Also, Internet is flooded with hardware flashers devices and software tools which are being used by the malicious persons for tampering and cloning the IMEIs of mobile devices [1]. These tampered mobile devices are then used for frauds, illegal acts, and anti- national activities and thereby cause a serious threat to security. The Law Enforcement Agencies and Telecom Enforcement agencies have a major task at their hands as to how to design innovative solutions and stop these rogue devices from entering any telecom network. In this article, we shall be discussing a new, hardware design based approach which shall be very effective against IMEI cloning and tampering. The paper discusses the use of life cycle based security of IMEI of the mobile device, implementation of challenge response system module in the hardware design, mirrored and majority principles for ensuring integrity of IMEI and its protection against tampering and cloning. The article also envisages the use of centralized EIR for secured telecom networks.

Keywords: Equipment identity register; International mobile equipment identity; IMEI cloning; IMEI tampering; IMEI protection; Life cycle security; Majority rule principle; Mirrored approach; Mobile device; Password challenge; Security.

I. INTRODUCTION

IMEI is a 15-digit numerical code consisting of first 8 TAC (Type Approval Code) digits, and then next 6 digits as serial number and 15th digit is the checksum digit as shown in Figure 1.

TAC	Serial Number	Checksum digit
1-8 (8 digits)	9-14(6 digits)	15 (1 digit) for Luhn Checksum

Figure 1: IMEI format and structure.

Luhn Algorithm [2] is used to find out whether the IMEI of the mobile device is valid or not. The mobile device user can find out the IMEI of the device by going to the settings of device. Also, there is USSD (Unstructured Supplementary Service Data) code defined for finding out the IMEI number of the mobile devices. The user can find IMEI by dialing *#06# on the mobile device. It is to be noted that if a mobile phone supports more than one SIM, it is mandatory for it to have that many number of IMEIs. For example, any dual SIM mobile phone must have two IMEIs.

1.1 Problem Statement

Malicious persons use hardware flashers devices and software tools [1] like IMEI changer tool, Patagonia etc. for tampering and cloning the IMEIs of the genuine mobile devices and these compromised devices pose a serious threat to security since they are used for carrying out frauds, illegal acts, and anti- national activities by malicious persons. In any telecom network which is having mobile devices with cloned or tampered IMEIs, the law enforcement and security agencies take much longer time to zero down the culprit and the course of their action often causes inconvenience to innocent persons who have the same IMEI of their mobile devices. So, we need to make sure that mobile devices with unique and valid IMEIs are allowed to enter our country's telecom network. Also, we need to make sure that only authorized persons can change the IMEI of any mobile device and that too in a defined lifecycle stage of the mobile device [3]. Since, tampering and cloning of IMEI, using the tools available on internet, is done using software, so we need to implement a hardware based solution in the mobile device which is robust enough against any software intrusion. Such hardware design must allow any change in IMEI only after due authentication of the user credentials using some kind of challenge response mechanism.

II. MOTIVATIONS AND PROPOSED HARDWARE SECURITY MODEL

The concepts of cryptography and separating IMEI into two parts and keeping one part in hardware design of the mobile device have been given as one of the solutions for protecting IMEI tampering. [4][5] These papers [4] [5] also concludes that Identification security must not depend on software security. The security architecture should be robust with respect to the tampering or modification of the software. Keeping this point in mind, we have designed a concept of hardware plus software security where the hardware design is robust enough against any tampering.

The approach, which we are proposing, is as follows:

We must store the IMEI of any mobile device in non-volatile memory i.e. EEPROM of the mobile device. Let us first define the concept of mirrored approach. In mirrored approach, we store a number at more than one place to ensure that we shall be able to retrieve the correct number from at least any one location in worst case, if the contents of all the other locations have been deleted, tampered or destroyed. Here we are using an advanced version of mirrored approach which shall be explained in the next section. We have also used the concept of majority rule i.e. using hardware logic we compare the values stored at all places and then finally decide the correct value as per majority. We shall use XOR gates and comparator based logic for implementing this majority rule concept in hardware logic. When a hacker with malicious intentions is reading the IMEI of any mobile device, he may play with the voltage levels of the EEPROM. The fluctuations in voltage might cause some stored binary values in memory to change/corrupt and thus we might get the wrong value of IMEI if we don't have this mirrored approach. Now, when we want to change the IMEI of our mobile device, it means we need to perform a write operation in the EEPROM. For this, we have implemented a Challenge Response System Module (CRSM) in our hardware design. This CRSM unit remains transparent when we are doing the read operation and it is activated only when we perform a write operation in EEPROM. So, during the read operation of IMEI, the integrity of IMEI is assured by the mirrored approach and majority rule principles.

2.1 IMEI Read Operation

The read operation and IMEI display of the screen of the mobile device takes place as per operations shown in Figure 2 and Figure 3.

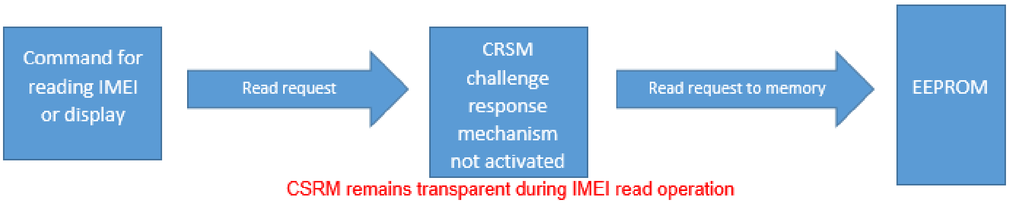
Command for reading IMEI or display
Read request
CRSM challenge response mechanism not activated
Read request to memory
EEPROM
CSRM remains transparent during IMEI read operation

Figure 2: Query generated by user for displaying IMEI of his mobile device.

In mirrored approach design, the IMEI, inverted IMEI, and IMEI right shift by 1 are stored at pre-defined locations in EEPROM. IMEI is a numeric value and 9's complement concept is used for finding out the inverted IMEI value. For example, if the IMEI is 864376021385686, its inverted IMEI will be 135623978614313. The right shift by 1 place version of IMEI is 686437602138568 (cyclic right shift).

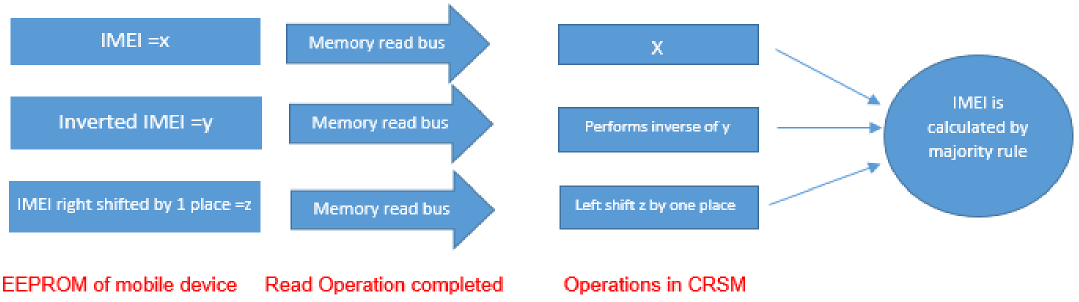

Figure 3: How to display the IMEI stored in your mobile device.

When we perform a read operation on our device for finding out the IMEI of our device, these values reach to CRSM via memory read bus and then the Inverted IMEI is again inverted using 9's complement to give the original IMEI. Similarly, right shifted IMEI value is shifted to left (cyclic left shift) by one place to give the original IMEI. Then a majority function is applied on these three values to display the final IMEI value. These three values of IMEI, Inverted IMEI and IMEI right shifted by 1 place are stored at different arrays of EEPROM. This is done to minimize the probability that the IMEI values will get corrupted simultaneously if any malicious person or hacker plays with the voltage levels of the EEPROM. In the analog design, we need to have the voltage sensors to ensure that if any hacker wants to corrupt these values by varying voltage of EEPROM, then the sensors don't allow it to happen and take the device to a secured state or cause the device to undergo reset and then shut it down [6].

2.2 IMEI Write/Change Operation

When we perform a write operation for changing/tampering the IMEI of the mobile device, the challenge response mechanism circuit in the CRSM gets activated. This challenge response mechanism is dependent on the life cycle of the mobile device show in Figure 4. Life cycle of any device moves from manufacturing stage to customer use and finally to failure analysis stage and the progress of the life cycle are in forward direction only. When the device is in manufacturing stage, say, the life cycle is LF_MANF and in this lifecycle, CRSM remains transparent and is not activated and the designers can change the IMEI for testing and validation purposes as many number of times as they want. The next life cycle is when the customer is using the mobile device, say LF_CUST. In lifecycle LF_CUST, when anyone tries to change the IMEI, CRSM gets activated and the user must provide the response for the password. Once the password and response matches, only then the write access is allowed. The last lifecycle stage is the failure analysis stage; say LF_FAL when the mobile comes back to the manufacturer with a catastrophic failure or defect and the manufacturer uses the mobile device for doing the root cause analysis of the failure. So, the life cycle always moves forward and is uni-directional only. Life cycle moves from LF_MANF -> LF_CUST -> LF_FAL

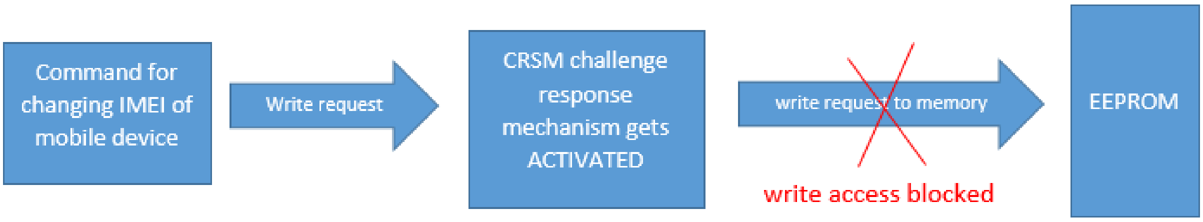
Command for changing IMEI of mobile device
Write request
CRSM challenge response mechanism gets ACTIVATED
write request to memory
write access blocked
EEPROM

Figure 4: Command given by user for changing IMEI of his mobile device.

When anyone performs a write operation for changing the IMEI of the mobile device, CRSM module gets activated and the write access to the memory is blocked. As a part of security feature, we have stored a binary valued password in the EEPROM. The user needs to insert a binary value called response/challenge which should match with the password stored in the memory for unblocking the write access to memory for changing the IMEI [7]. The length of this password and response can be from 32 bit to 256 bits depending on the password strength and protection we are aiming for. The flow chart for the password challenge and response mechanism is shown in Figure 5. We have done the simulations for password and response lengths 32, 64,128, 256 bits and the results are shown in Figure 6. As we increase the length of the password and response, the probability that a malicious person can match and qualify this mechanism decreases and IMEI security is enhanced. But the hardware logic takes more time in comparison, we need more number of XOR gates for comparison and we need more space in memory for storing the password. So, keeping these trade-offs in mind, we can employ 32 to 256-bit password challenge and response mechanism. We can even further enhance the integrity of the stored password using the mirrored approach for storing the password at multiple places in memory. We can also limit the number of times a user can input the response so that the malicious persons don't get enough time and attempts to match the password.

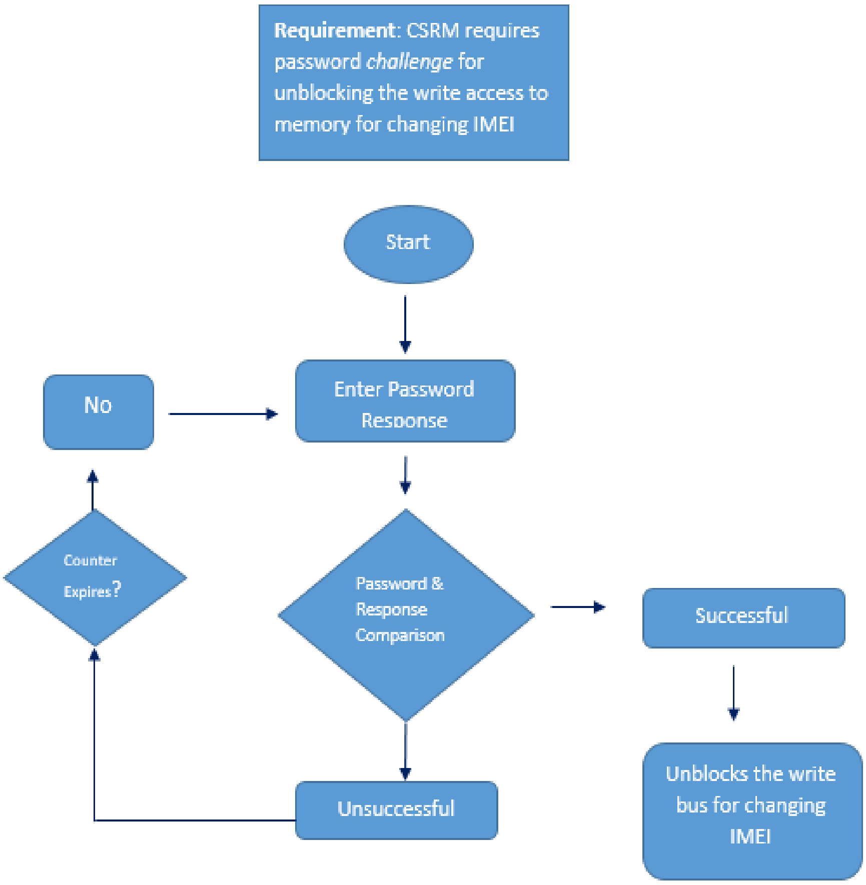
Requirement: CSRM requires password *challenge* for unblocking the write access to memory for changing IMEI
Start
Enter Password Response
No
Counter Expires?
Password & Response Comparison
Successful
Unblocks the write bus for changing IMEI
Unsuccessful

Figure 5: Flow chart for Password challenge and Response mechanism for changing IMEI.

The number of attempts can be limited using a counter based approach. When, the counter expires, the write bus is permanently blocked and thus anyone cannot change the IMEI. The read bus remains available for reading the IMEI of the mobile device. When the mobile manufactures are storing the password in memory, they need to make sure that combinations of all 1s and all 0s are avoided as passwords and such combinations must be added to exception list when inputs are given to pseudo random binary numbers generator tools [8].

III. SIMPLIFIED CODE AND RESULTS

A very simple version for the password challenge and response mechanism is as follows:

```
`Define IMEI_WIDTH 256 // can be 32, 64, 128, 256 bits length
`Define CHALLENGE_COUNT 32 // number of attempts for comparison
class imei_drv;
 rand bit [`IMEI_WIDTH-1:0] imei_challenge;
 rand bit [`IMEI_WIDTH-1:0] imei_password;
 int counter;
 bit locked, hacked;
task imei_task;
 begin
   randomize(imei_password);
   $display("IMEI:%x",IMEI generated and stored in memory);
while(!locked & !hacked)
begin
#1;
randomize(imei_challenge);
$display("IMEI_challenge:%x",imei_challenge);
if(imei_challenge != imei_password)
counter ++;
else begin
$display("IMEI Password Matched !!! write access unblocked !!"); //
hacked = 1;
end
if(counter == `CHALLENGE_COUNT) begin
$display("Device locked, write access blocked");
locked =1;
end
end
end
endtask
endclass
program imei;
  imei_drv drv_inst = new();
  initial begin
  drv_inst.imei_task;
  end
endprogram
```

The output of the code with 32 bits IMEI password and challenge and 32 attempts counter is as follows in Table 1:

IMEI generated and stored in memory:	**Response Attempt Counter**	21dbe4ac
IMEI_challenge:	1	99845814
IMEI_challenge:	2	01eb90df
IMEI_challenge:	3	366c4159
IMEI_challenge:	4	9a1d9940
IMEI_challenge:	5	4932bb32
IMEI_challenge:	6	b10acf0c
IMEI_challenge:	7	aaa1e714
IMEI_challenge:	8	b58b7fcf
IMEI_challenge:	9	7c7e9ae3
IMEI_challenge:	10	c1da7c5b
IMEI_challenge:	11	92449dad
IMEI_challenge:	12	00fe718a
IMEI_challenge:	13	61b492eb
IMEI_challenge:	14	928c4087
IMEI_challenge:	15	2d8bac43
IMEI_challenge:	16	74c11e36
IMEI_challenge:	17	f2f5053c
IMEI_challenge:	18	f9bc9256
IMEI_challenge:	19	47e0455e
IMEI_challenge:	20	9184fa7a
IMEI_challenge:	21	9788e7c3
IMEI_challenge:	22	deb1b7d2
IMEI_challenge:	23	3be87a72
IMEI_challenge:	24	b9c02ac1
IMEI_challenge:	25	5984476b
IMEI_challenge:	26	c95337e7
IMEI_challenge:	27	982e70b8
IMEI_challenge:	28	1a841540
IMEI_challenge:	29	596a6f06
IMEI_challenge:	30	3cb688b3
IMEI_challenge:	31	ac5547f4
IMEI_challenge:	32	147e2bb0
		Device locked, write access blocked
		Simulation complete via implicit call to $finish (1) at time 32 NS + 1

Table 1: Output of Password Response mechanism with IMEI password length 32 bits & counter attempts = 32.

The overall times and probabilities of password challenge and response mechanism for password match with respect to length of passwords are shown in Table2 and Fig6 respectively. We have assumed that the mobile device and attached hacker device is using 2 GHz clock. The time taken for password match has been found out in worst case scenario. It is evident from Table2 that as we increase the length of password, the time taken for password match increases exponentially. The counter which we have implemented in the design limits the number of times you can run any algorithm for guessing the password and thus reduces any overhead on the hardware logic.

Length of the IMIE password (in bits)	Number of possible combinations	Worst time taken for password match assuming fastest search @ 2GHz clock	Probability of password match
4	16	8 ns	0.0625
8	256	128 ns	0.00390625
16	65536	32768 ns	0.0000152587890625
32	4294967296	2147483648 ns	2.32830643653869E-10
64	18446744073709500000	9223372036854770000 ns	5.42101086242752E-20
128	3.40282366920938E+38	1.70141183460469E+38 ns	2.93873587705571E-39
256	1.15792089237316E+77	5.7896044618658E+76 ns	8.63613655509444E-78

Table 2: Time taken for password match with respect to variable IMEI password lengths.

In Figure 6, Series 1 legend shows the length of password and series2 legend shows the probability of password match.

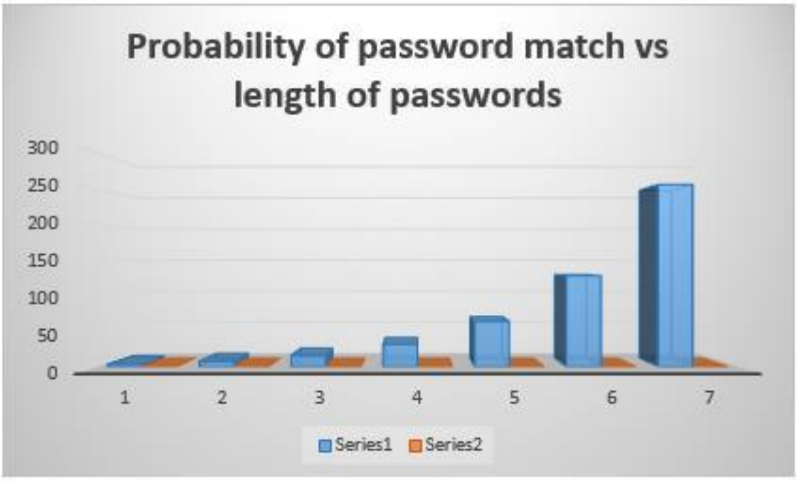

Figure 6: Probability of password match with respect to variable IMEI password lengths.

From Table 2 and Figure 6, we conclude that as the password length increases, it becomes more and more difficult for any malicious person or hacker to guess the password and break the IMEI password and security mechanism.

IV. IMPLEMENTATION OF EIR IN TELECOM NETWORKS

The telecom standards and technologies enable us to implement EIR module in telecom networks for storing the IMEI of all the mobile devices being used in that telecom network. But this implementation is not mandatory in all the countries. IMEI is the identity of the person who is using that mobile device. So, EIR implementation should be made mandatory so that the malicious persons who are using mobile devices for illegal and anti-national activities can be tracked. Furthermore, using this EIR rogue mobile devices with cloned or tampered IMEI can be traced and blocked from entering telecom networks and made non-functional. This EIR repository database should be shared with GSM association in a centralized manner. GSMA allots IMEI to all the mobile manufacturers and sharing of this database will make sure that only mobile devices with valid IMEIs are being used worldwide.

V. CONCLUSION

This proposed hardware design for IMEI security and simulation results of the tests carried out on it will make sure that IMEI cloning and tampering becomes next to impossible and only mobile devices with valid IMEIs are used in any telecom networks across the globe. Furthermore, the mandatory implementation of Centralized Equipment Identity Register in every telecom network will help the Law Enforcement and Telecom Enforcement agencies to track the malicious persons who use mobile devices for carrying out illegal and anti-national activities.

VI. REFERENCES

1. Iclarified, how to change your iPhone IMEI with ZiPhone window 2008.
2. J Naveen, S Chetan, Looking at Mobile Phones with a Different Perspective. Communications Today 2016.
3. GSMA/EICTA, Security Principles Related to Handset Theft 3.0.0 Information system communication technologies consumer electronics 2005: 1-10.
4. G Igor, General Requirements and Security Architecture for Mobile Phone Anti-Cloning Measures. IEEE EUROCON 2015.
5. Z Sergiy, Z Igor, Mobile Phone Multi-Factor Authentication with Robustness of Clone Detection. Modern Problems of Radio Engineering Telecommunications and Computer Science TCSET 2016: 23-26.
6. P La Mariantonietta, M Fabio, et al. A Survey on Security for Mobile Devices. IEEE communications surveys and tutorials 2013; 15: 446-471.
7. A Abdullahi Arabo, P Bernardi, Mobile Malware and Smart Device Security: Trends, Challenges and Solutions. Control Systems and Computer Science 2013.
8. LFJ Alexandre, G David, et al. Impact on Network Quality and a New Method to Identify an Unlicensed IMEI in the Network. IEEE Communications Magazine 2014; 52: 90-96.

VII. BIOGRAPHY

Naveen Jakhar is an Officer of Indian Telecommunication Service. He has received B.E. degree (honours) in Electronics and Communication Engineering from Netaji Subhas Institute of Technology, University of Delhi in year 2012. His areas of interest are: Cyber Security, Mobile Communication technologies, Data communication, Satellite Communications, Transmission technologies.

Prior to joining I.T.S, he has worked with Freescale Semiconductors for a period of 3 and half years in the capacity of VLSI Design Engineer with expertise in domains : Clocking and Reset Architectures of SoCs, Low Power Verification, Different modes of Operations of SoCs, CPF & PGLS for Low Power Verification, Verification of Cortex ARM53 complex, GIC - Generic Interrupt Controller, CCI - Cache Coherency Interconnect and Cache Coherency at SoC level, Caches Verification, AMBA AHB, AXI and ACE Protocols.

He has authored 11 articles and white papers in International magazines and journals.