1. **List of team members**
   Amit Singh

2. **Title of project**
   "All Programmable System on Chip Security"

3. **Introduction & Motivation**
   All Programmable SoCs load their *application* image files from one of multiple sources like SD card, network & so on. The image files of these systems are susceptible to multiple attacks (including both by physical & remote access). SoC chips are favored across *electronics design industry* for their flexibility & are consistently gaining popularity for their applications in consumer, medical, defense & industrial applications. This originates an urgent need to standardize the security protocols followed in manufacturing, designing & operations of these chips.
   Large corporations like Microsemi & Xilinx have taken the initiative to design & market "All programmable SoCs" with their own approach towards security protocols. This project will allow analysis of the security paradigm these *corporations* have taken. Since, these products are still in evolving phase it is worth pointing out strengths, weaknesses & possible future trends for these products. In present studies, there has been a generic opinion on SoC security. Although, one IEEE paper is been published on Xilinx Zynq SoC in October 2014, there has not been much study on Microsemi Smartfusion2 SoC. By analyzing these two SoCs on common parameters, we should be able to determine if sufficient amount of security is achievable using one of the two SoCs for a given application.
   As we know in current generation every networked device is accessible for an attack, which in turn motivates system designers to ensure if they are able to meet the security requirements of posing threats. Thus, an insight study related to security apparatus of SoCs could make non-trivial impact in rapid prototyping & also contribute towards a universal security mechanism/standard.

4. **List of alternative solutions**
   Microsemi SmartFusion2 & Xilinx Zynq-7000 All programmable SoCs could pose as alternative to each other depending on the strength of their security mechanisms.

5. **Tentative list of Evaluation Criteria**
   - Key generation, exchange & management techniques
   - Security services at different stages (while in operation) in product life cycle
   - Root of trust
   - Countermeasures (or action by system) in case of a breach

6. **Mutual dependencies among various evaluation criteria**
   Yes, mutual dependencies exist among different evaluation criteria. For an instance, strength of the *cryptosystem* could vary from estimation if root of trust is missing.

7. **Detailed descriptions of problems/hypothesis:**

   Security mechanism while *key generation*, *loading* & when system is *in operation*: Key secrecy needs to be ensured at all times. How is it exchanged? Where is it stored? & how it can be accessed? Are important questions that need to be answered.

   Countermeasures in case of partial breach: In case of breach due to some loopholes in application, detection of attack & response to it remains a major challenge

   Ease of using security features by designers: Following the security protocols without greater loss in maneuverability of designer while designing application, needs to be met.

   Previous studies on SoCs: The available literature on FPGA security measures resource consumption, code size & difference in time, while literature on SoC security involves mapping of security libraries, adding trusted modules onto SoC & etc. The proposed project would analyze the basic cryptographic implementations (like Hash functions, key arrangement, PKC mechanism & so on) to understand if any additional security implementation is required.

8. **Tentative list of questions**

   1) Is key secured?
   2) Is there any possibility of breach in root of trust?
   3) Are in-built countermeasures sufficient in case of breach in security?
   4) Which attacks can be prevented, how?

9. **Procedure/experiments used for verifying the results**

   1) Step by step analysis of given security specifications in respective SoCs.
   2) Reports provided by *respective corporations* will form the majority source for analysis.
   3) Reliable findings of *independent entities* could be considered as a source of information.
   4) Considering possible *threat scenarios* & determining the impact of *built-in counters* for these threats.

10. **Time Schedule**

    Oct 26-27: Overview of security mechanisms (involves identification of security blocks)

    Nov 9-10: Report on analysis of key management (involves determining the impact on security due to specific key loading methods)

    Nov 23-24: Report on analysis of specific security mechanisms, countermeasures & flexibility in implementation (involves analysis of stages in secure boot process, tackling an attack & ease of use)

    Dec 7-8: Report on analysis of root of trust & final report

11. **List of possible areas, where the specification can change depending on the progress of the project**

    1) Different approach by both corporations in implementing their security mechanism could lead to some changes in evaluation criteria.
    2) Any new updated review on this topic, might force some addition/modification in the specifications.

12. Tentative table of contents of final report

1) Introduction
2) Overview of security mechanisms
3) Cryptosystems & key management
4) Security services at various stages of boot process
5) Root of trust
6) Countermeasures
7) Conclusion & future scope

References

13. List of Literature

[1] P. Trott, “Secure Boot-FPGA Anti-Tamper Solution DPA-safe SRAM FPGA configuration with Smartfusion2,” Microsemi, Aliso Viejo, CA, Tech paper, 2014.

[2] L. Sanders, “Secure Boot in the Zynq-7000 All Programmable SoC,” Xilinx Inc., San Jose, CA, White Paper WP426 (v1.0), April 5, 2013.

[3] SmartFusion2 as a Hardware Root-of-Trust. (n.d.). Microsemi Corp. [Online]. Available: http://www.microsemi.com/products/fpga-soc/security/secure-boot-fpga. Accessed Sept. 21, 2015

[4] E. Peterson, “Developing Tamper Resistant Designs with Xilinx Virtex-6 & 7 Series FPGAs,” Xilinx Inc., San Jose, CA, App. Note: Virtex-6 & 7 Series FPGAs, & Zynq-700 AP SOCs, Oct. 2013.

[5] O. Khalid, C. Rolfes, A. Ibing, “On Implementing Trusted Boot for Embedded Systems,” Fraunhofer Research Institution for Applied & Integrated Security, Munich, Germany, IEEE Symposium on HOST, 2013.

[6] M. M. Parelkar, “FPGA Security – Bitstream Authentication,” George Mason University, Fairfax, VA, Tech. Paper, Date N.A.

[7] D. Schellekens, P. Tuyls, & B. Preneel, “Embedded Trusted Computing with Authenticated Non-volatile Memory,” Katholieke Universiteit Leuven, ESAT-SCD/COSIC, Belgium, Tech paper, 2008.

[8] A. Ukil, J. Sen, S. Koilakonda, “Embedded Security for Internet of Things,” TCS, Kolkata, India, Date N.A.

[9] S. Babar, A. Stango, N. Prasad, J. Sen & R. Prasad. (2011 Jan.) Proposed Embedded Security Framework for Internet of Things. [Online]. Available: http://www.researchgate.net/publication/252013823_Proposed_embedded_security_framework_for_Internet_of_Things_(IoT)

[10] “Cryptoflex,” Class notes for CS 198, Computer Science, University of California Los Angeles Winter 2011.

Additional literature:
Manuals from Microsemi & Xilinx