A Low-cost and High-performance SoC Design for OMA DRM2 Applications

Yehua GU, Xiaoyang ZENG*, Jun HAN, Jia ZHAO

State Key Lab of ASIC & System
Fudan University
Shanghai 201203, China
{042052019@fudan.edu.cn, xyzeng@fudan.edu.cn}

Abstract- A SoC design for applications of OMA DRM 2 Agent in mobile phones is presented in this paper, which has been verified by Altera Stratix EP1S80B956C6 FPGA development board. Several design aspects, which include an embedded 32-bits RISC CPU and AMBA™ bus system, a DRM Agent accelerator, a high-performance TRNG, several interfaces and reasonable hardware/software partition, making it very efficient for the OMA DRM 2 application. Based on SMIC 0.25μm standard CMOS technology, the proposed SoC platform can work under the frequency of about 76MHz, and the core circuit is 112k gates, making it suitable for low-cost design. Besides, memory protection unit is added to enhance the security. Therefore, the proposed SoC platform has a fine potential in application.

I. INTRODUCTION

With the popularization of mobile phones, mobile internet is becoming an important channel for digital content distribution. To protect the intellectual property rights of the authors and issuers, the digital content, such as polyphonic ring tones, mp3 audio files, games and so on, must be consumed in a controlled way. Therefore, the need for mobile digital right management (DRM) solution is intensified.

The Open Mobile Alliance (OMA) has finished their work on the second version of DRM specification [1] in 2004. With respect to the first version, OMA DRM 2 provides a higher level of security mechanisms. Its security lies on the Public Key Cryptosystem (PKC) for key distribution and symmetric encryption algorithms for content protection. There are four actors in the DRM system [2]: Right Issuer (RI), Content Issuer (CI), Certification Authority (CA) and DRM Agent. They interact with each other in order to provide access to Right Object (RO) to the end-user, while how to obtain Content Object (CO) is not defined. DRM Agent has to perform four steps to consume protected content:

1. Establishing Trust
2. Acquisition of RO
3. Installation
4. Consumption

Both in step (1) and step (2), DRM Agent needs to make a signature on sent data and verify the certificate from RI. After step (2), RO and CO have been received separately and stored in DRM Agent. According to Daniel Thull and Roberto Sannino’s work [2], the following task has to perform in the Installation and Consumption phase, as shown in Fig.1. Firstly, DRM Agent decrypts C1 (the first 1024 bits of C which is contained in the RO) using its private key and obtains Z. Applying the KDF2 function to Z yields K_{KEK}. Then decrypts C2 using the key K_{KEK} and obtains two keys, K_{MAC} and K_{REK}. K_{MAC} is used to check the integrity of RO and K_{REK} is the key for the decryption of K_{CEK}. Both K_{MAC} and K_{REK} are encrypted using a device-generated key K_{dev} and obtained C_{2DEV}, which is then stored in RAM. Secondly, in the phase of consumption, DRM Agent decrypts C_{2DEV} using K_{dev} and verifies RO integrity by checking its MAC and CO integrity by calculating its Hash value. After that, decrypts C3 and obtains K_{CEK} which is used to decrypt CO.

![Diagram](https://example.com/diagram.png)

**Fig.1. Installation and Consumption**

Therefore, DRM Agent does such work as following:

1. Signature and verification
2. RSA decryption
3. AES WRAP en/decryption
4. AES CBC decryption
5. Hash value calculation, both SHA-1 and HMAC
6. KDF2 as key derivation function

The other part of this paper is organized as follows. Section 2 makes reasonable hardware/software partition. Section 3 gives a low-cost and high-performance SoC implementation, meanwhile the security issue is considered. Section 4 shows the experimental results. Section 5 concludes this paper.
II. HARDWARE/SOFTWARE PARTITION

According to the specification of OMA DRM 2, DRM Agent performs the task of calculations and communications. Obviously, Platform-based SoC is an efficient solution for DRM Agent [3], which is easy to carry out complicated protocols under the control of CPU. The most important features of DRM Agent are speed, area and power consumption. Three cryptographic algorithms: AES, RSA and SHA-1, can be accelerated by implementation in hardware, as shown in Table I [2]. The complex flow-control can be realized in software.

In this paper, the AES, RSA, SHA-1 hardware accelerators are combined to one named as DRM Agent accelerator.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Software [cycles]</th>
<th>Hardware [cycles]</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA Encryption</td>
<td>2160000/1024 bit</td>
<td>10000/1024 bit</td>
</tr>
<tr>
<td>RSA Decryption</td>
<td>37740000/1024 bit</td>
<td>260000/1024 bit</td>
</tr>
<tr>
<td>SHA-1</td>
<td>400/128 bit</td>
<td>20/128 bit</td>
</tr>
<tr>
<td>AES Encryption</td>
<td>360+830/128 bit</td>
<td>10/128 bit</td>
</tr>
<tr>
<td>AES Decryption</td>
<td>950+830/128 bit</td>
<td>10+10/128 bit</td>
</tr>
</tbody>
</table>

III. SOC IMPLEMENTATION

A Overall SoC Architecture

Traditional information security SoC architecture, which consists of an embedded CPU and several isolated coprocessors, optional DMA controller, memories, and some other assistant modules, as shown in Fig.2. However, this solution may have a deficiency. For example, the attacker can get sensitive information by analysing the bus data during the procedure of working. And it takes more time to exchange data because the coprocessors are isolated. Further more, the power consumption will dramatically increase as a large mount of calculation is left to CPU, which correspondingly shorts the battery life in mobile device.

The proposed new SoC architecture is shown in Fig.3. It is based on AMBA™ bus, which contains a DRM Agent accelerator with only one AHB interface to the system bus. RAM and ROM are protected by special protection unit.

All the calculations in DRM Agent are restricted in an area named “Secure Zone” to prevent from attack. The system bus AHB only transports the content of RO and CO, which are stored in the buffer RAM. The device private key is stored in the flash ROM by the foundry or user in a special way. Visits to protected area in RAM and ROM from AHB will be forbidden.

The AES, RSA and SHA-1 accelerators are connected directly in DRM Agent accelerator. Therefore reduce the time for load and store. Furthermore a dual-port buffering RAM is used, which supports reading and writing different address at a clock.

In the traditional SoC architecture, AES, RSA, SHA-1 modules have their own AHB interfaces and buffer RAM. As the DRM Agent works in a sequential fashion, these modules can share their AHB interface and buffer RAM to reduce the area.

As we know, CPU is one of the most power consumption units. In this design, DRM Agent accelerator carries out most of the calculations while CPU enters its suspend mode until an interrupt comes.

B DRM Agent Accelerator

Fig.4 shows the overall architecture of DRM Agent accelerator which consists of following several important sub-modules: the re-configurable and scalable public-key RSA accelerator [4], AES and SHA-1 accelerators [5,6], TRNG (Truly Random Number Generator) and some other modules.

The AES, RSA and SHA-1 modules have been optimized for their usage. Configuration and Status registers receive command from CPU, send start and control signals to the selected modules, meanwhile reflect the state of DRM Agent accelerator.

TRNG is an analogue module and mainly used for key generation. In this design, a thermal noise based scheme is adopted. And the TRNG has passed the strict test with the international standards such as FIPS140-1 and NIST SP800-22.
The core arithmetic of RSA is modular exponentiation, which can be accomplished by a sequence of modular multiplication. Therefore, the performance of modular multiplier influences RSA module greatly. This paper proposes an improved calculation cell structure in modular multiplier, according to Tenc's MWR256MM algorithm [12]. As shown in Fig.5, it includes such parts as Booth unit, Encode unit, two carry-save adders and so on. The Booth unit, which is commonly used in long integer multiplier, is adopted here for reducing the number of partial products. The Encode unit is employed to avoid multiplication and inverse. Carry save adder is effectively decreases the delay in the datapath. Consequently the output of the calculation cell consists of two parts: SS and SC. A 32-bit carry look-ahead adder is needed to get the sum of SS and SC.

Most hardware implementations of AES algorithm do the 128-bit calculation of every round in parallel. Although these designs achieve high throughput, their hardware cost are too high. It is known that AES algorithm is block calculation, so the 128-bit calculation of every round can be split into four 32-bit computations. If the computation of one 32-bit data is done in one clock, it will take four clocks to complete one round calculation. However, this approach requires the complicated MixColumn and SubByte modules to be connected in series, which will increase the critical path and decrease the chip speed. In order to overcome the problems above, this paper applies a pipelined structure to finish the calculation of every round. This pipelined structure achieves high calculation efficiency with low hardware cost. As demonstrated in Fig.6, interpolating one 32-bit register (Reg_i) and adding one extra clock is enough to enable this scheme to finish the calculation of one round in five clocks. As a result, the length of the critical path is cut down and the chip speed is raised.

In SHA-1 module, this paper modified its datapath for high speed at the cost of extending the critical path [13], as shown in Fig.7. It performs two iterations in one cycle, therefore cut down half of the execution time.

\[
E_t = E_{t-2} + S^{30}(B_{t-2}) + f_t(S^{5}(A_{t-2})) + W_{t-1} + K_{t-1}
\]

\[
D_t = D_{t-2} + S^{30}(B_{t-2}) + f_t(S^{5}(A_{t-2})) + W_{t-2} + K_t
\]
C Memory Protection Unit

Fig. 8 shows the memory protection unit architecture, which is used to protect the content in memory from attacks. Memory is divided into three districts. The first district can only be written, where DRM Agent private key and some sensitive information is loaded during the calculation. The second district can only be read, where decrypted data is stored. The third district can be read and written, where unimportant data is stored. Control Center in this unit sends commands according to user ID and accessing address. Common user’s visits to protected area in memory from AHB will be denied.

![Memory protection unit architecture](image)

Fig. 8. Memory protection unit architecture

IV. EXPERIMENTAL RESULTS

The proposed SoC architecture is described in RTL, and synthesized to gate-level with Synopsys Design Compiler based on SMIC 0.25μm standard CMOS technology. The synthesis results show that it can operate under the frequency of 76 MHz, and the total area is about 112K gates (RAM for Content Object excluded).

Table II shows the gate counts for the main modules in this SoC platform. For high-performance, a 32-bits RISC CPU is used. Other modules are optimized for their objects.

Table III gives the comparison result with other designs, which shows that the DRM Agent accelerator achieves the object of both high-performance and low-cost.

V. CONCLUSION

This paper presents a new SoC solution for DRM Agent, which includes an embedded 32-bit RISC CPU and AMBA bus system, a DRM Agent accelerator, a high-performance TRNG and some other low-cost modules. And the DRM Agent accelerator achieves the goal of both high-performance and low-cost. Therefore, it is very efficient for applications in mobile phones.

REFERENCES