AES Algorithm Applied on Security Protocol of RFID

Liquan Han, Fang Yuan, Yuqiang Jiang
College of Computer Science and Engineering Changchun University of Technology Changchun P.R. China

ABSTRACT: The radio frequency identification (RFID) system may suffer from serious threats that may result in security problems, which becomes a critical issue for RFID systems and applications. As a consequence, a security protocol for RFID tags is needed to ensure privacy and authentication between each tag and their reader. In order to achieve this purpose, this paper proposes a new encryption security authentication protocol, which Based on the research of The Advanced Encryption Standard, AES algorithm is applied to RFID system to ensure the safety of data transmission. The paper also discusses the hardware implementation of AES algorithm detailed, and uses the basic structure of feedback mode to achieve encryption and decryption operation. The top-down approach to the design of hardware module work. The completion of the hardware describe language of all the module functionality test, And meet the requirements of the protocol RFID.

1 INTRODUCTION
RFID is a technology for automated identification of objects and people with electromagnetic fields. Conceptually, RFID is similar to a bar-code system, but its wireless communication allows significant qualitative advances. The reader need not have touches to the tag and interrogates multiple tags at the same time. The tag can store many more bits of information. There are various applications for low-cost and low power tags such as logistics, point-of-sales, animal identification, item management, and so on. Thanks to advances in the capabilities of tags, drastic decreases in the cost of RFID system, and many efforts to adapt it in the real world, RFID system seems to replace optical bar-code and proliferate in the near future. However, the radio communications between RFID tag and readers raise a number of security issues. Basically, RFID tags send their identifier without more security verification when they are powered by electromagnetic waves from[1]

In the passive radio frequency interfaces, frequency range at 13.56 MHz the maximum means current consumption without reducing the operation range of the tags is 15 µA. Due to the limited available chip performance, the limited power and the limited time, an algorithm is allowed to execute, the selection of appropriate security algorithms and protocols are very crucial. The main purpose of authentication technology is to guarantee the accuracy of the tag reader to identify the ID of the reader, and prevent the storage data information without permission is illegal to read and malicious tampering, at the same time also must ensure that the Reader correct identification Tag the Tag's identity, to prevent the Reader read the fake data information mutual authentication between Reader and Tag. This article applies the international standard IS09798-2 certification process[11]. This security protocol for passive RFID tags based on AES is according to three mutual authentication protocol[2].

2 AES ALGORITHM USED IN THE PROTOCOL
The AES module requires only a little chip area and power. The AES algorithm is a symmetric block cipher with a variable block length and a variable key length. Therefore AES is a flexible algorithm for hardware implementations. This flexibility of the AES algorithm was intended by its creators. Efficient implementations are possible on 32-bit, 64-bit, 128-bit and 256-bit, platforms.[8] AES module for RFID tags regard low size and low power consumption requirements. What we have designed is used 128-bit platforms [13].

Each common data block of AES is modified by several predefined rounds of processing, where each round involves four functional steps. As figure 1 indicates, the four steps in each round of data encryption are called SubBytes, ShiftRows, MixColumns, and AddRoundKey [7]. Each small squares in the graphic represents a byte, or 8 bit, and each matrix has a total of 16 or 128 - bit small squares. the standard of a total of 128
- bit AES encryption process needs to be done for 10 rounds operation, and the first round only AddRoundKey operation performed operation, the 10 rounds does not run MixColumns operation [6]. The whole process of AES is shown in Fig.1. A small box represents a byte equal 8 bit. And there is no MixColumns step in the 10th round.

2.1 **SubBytes substitutes each byte of the State**

This operation is non-linear. It is often implemented as a table look-up [7]. Nonlinear change single byte substitution. SubBytes only handle each byte to the bytes in the finite field GF (28) on the multiplication inverse, ”0” is mapped to itself, which form $a \in GF(28)$ to $\beta \in GF(28)$. To get $a \cdot \beta = \beta \cdot a = 1 \mod (x^8+x^4+x^2+x+1)$.

\[
\begin{array}{c|c}
 y_1 & 11111000 \\
 y_2 & 01111100 \\
 y_3 & 00111110 \\
 y_4 & 00011111 \\
 y_5 & 10001111 \\
 y_6 & 11000111 \\
 y_7 & 11100011 \\
 y_8 & 11110001 \\
\end{array}
\begin{array}{c}
 x_1 \\
 x_2 \\
 x_3 \\
 x_4 \\
 x_5 \\
 x_6 \\
 x_7 \\
 x_8 \\
\end{array}
\begin{array}{c}
 0 \\
 1 \\
 0 \\
 0 \\
 0 \\
 1 \\
 1 \\
 0 \\
\end{array}
\]

\[y_i = x_i + x(i+4) \mod 8 + x(i+6) \mod 8 + x(i+7) \mod 8 + c_i,\]

This pattern is too complicated to realize in hardware. So we usually use S-box to substitute. S-Box is a 16 * 16 fixed array and independent for each byte.

2.2 **ShiftRows rotates each row of the state by an offset.**

The actual value of the offset equals the row index, e.g. the first row is not rotated at all; the last row is rotated three bytes to the left.

2.3 **MixColumns transforms columns of the State**

It is a multiplication by a constant polynomial in an extension field of 128-bit. The MixColumns multiplies the input polynomial by a constant polynomial $a(x)$, given by As shown in equation, the MixColumns operation for one column is written as output bytes $b_i$ of MixColumns are calculated by the same function just the order of the input column bytes $a_i$ differs [4].

\[b(x) = (03x^3+01x^2+01x+ 02) \cdot a(x) \mod (x^4 + 1)\]

So we could get available matrix form below:

\[B(x) = c(x) \otimes A(x)\]

\[
\begin{array}{c|c|c|c|c|c|c}
 A_{00} & A_{01} & A_{02} & A_{03} & B_{00} & B_{01} & B_{02} \\
 A_{10} & A_{11} & A_{12} & A_{13} & B_{10} & B_{11} & B_{12} \\
 A_{20} & A_{21} & A_{22} & A_{23} & B_{20} & B_{21} & B_{22} \\
 A_{30} & A_{31} & A_{32} & A_{33} & B_{30} & B_{31} & B_{32} \\
\end{array}
\]

\[\times (00)\]

AddRoundKey It combines the 128-bit State with a 128-bit round key by adding corresponding bits mod 2. This transformation corresponds to a XOR-operation of the State and the round key [9].

3 **AES ALGORITHM INTEGRATED SIMULATION MODULE**

AES hardware encryption circuit in this system not only needs to be done on AES-128bit encryption, also need to consider the sequence of hardware circuit, parameters such as power consumption. Because the particularity of RFID tags is limited, the requirements of passive low power and small area in
the design of the module, the need to consider low area of low power consumption as much as possible.

3.1 AES circuit top structure

Port on the tops of AES circuit diagram and the port list are shown in figure 5 and table 1, AES_ED is for encryption selection signal; AES_START is for open signal, AES_DONE is the sign of complete encryption system signals, and the effective AES_DOUT represent encrypt or decrypt data.

![Figure 5. AES top port.](image)

Table 1. Parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>BIT</th>
<th>I/O</th>
<th>Instructions</th>
<th>Name</th>
<th>BIT</th>
<th>I/O</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>rst_n</td>
<td>1 bit</td>
<td>Input</td>
<td>outside reset signal</td>
<td>aes_key</td>
<td>128 bit</td>
<td>Input</td>
<td>private key</td>
</tr>
<tr>
<td>clk</td>
<td>1bit</td>
<td>Input</td>
<td>outside clock signal</td>
<td>aes_dout</td>
<td>128bit</td>
<td>Output</td>
<td>En/Decryp output</td>
</tr>
<tr>
<td>aes_ed</td>
<td>1 bit</td>
<td>Input</td>
<td>en/decryp choice</td>
<td>aes_start</td>
<td>1 bit</td>
<td>Input</td>
<td>beginning signal</td>
</tr>
<tr>
<td>aes_din</td>
<td>128 bit</td>
<td>Input</td>
<td>data input</td>
<td>aes_done</td>
<td>1 bit</td>
<td>Output</td>
<td>finished signal</td>
</tr>
</tbody>
</table>

The primitive functions SubByte and ShiftRow are based on byte-oriented arithmetic, and AddRoundKey is a simple 128-bitwise XOR operation. Their operating order is not important because SubByte operates on one single byte, and ShiftRow reorders byte data without changing them. It can save clock cycles which are consumed to operate a separate AddRoundKey function. The efficiency of a AES cryptographic hardware in terms of circuit area, power consumption, and throughput is mainly determined by the data [6].

AES module of this system can use the same set of hardware encryption to complete the decryption process, the structure is shown in figure 1. In this figure the En/Decryp Engine is the core computation, the part and KeyExpansion encryption is the same operation, by selecting the signal to SubBytes MixColumns computing to complete the operation requirements [13].

In order to optimize our AES circuit, resource sharing in the data path is fully employed. We also used some low power circuit design technologies. At the gate level, gated clock is used to reduce switching activity of latches and flip-flops. Data gating is used to decrease unwanted switching in combinational logic blocks. At the architectural level, we try to optimize the data path by reordering and integrating the functional steps of AES algorithm [9]. The main goal of these low power circuit design methods are to reduce dynamic power consumption by reducing unwanted switching activities. For our low power AES circuit, the first step was to find a minimal architecture. This part was done by hand. A set of key components thus obtained. Components of AES circuit then designed and applied aforementioned low power techniques to each component. It unrolls only one round operation, and loops data through this round operation until the entire encryption is completed. There are several key components for AES circuit, a controller, data and key memory, S-box, and data path.
3.2 Authentication control circuit part

As shown in the diagram, when the master state machine jumps to the ACTIVE module, the module is started and waits to receive data from the reader. R is obtained by AES decryption, and at the same time, the interface block of storage, EEPROM, locates R, if they are the same, then the reader is legal, then continue the next step. The random number and the application ID are encrypted with AES and transmitted. The encryption results are saved again and refreshed in the chip.

Figure 7. Process of authentication.

Above flow chart clearly describes the RFID successfully passed the mutual authentication under the active statement. This flow chart illustrates the communication in the data encryption process. After successful authentication and receiving data, the block reads EEPROM to get AES_KEY in specific areas, decrypt to get order, and sends to the EEPROM interface module and does the corresponding processing. If reading signal is received, reading data is directly sent to the AES module, encrypted with AES - 128bit, and sent out again.

4 SYNTHESIS AND SIMULATION RESULTS

This design is only applied for the simulation of AES-128bit encryption function, and the data is typical, including some boundary data. According to these encryption results, we can compare the value of the encryption whether equals with the standard values, to verify the AES module function is correct.

As is showed in figure below, this is a typical AES-128 bits encrypted waveform sequence diagram. From the diagram, when aes_strt signal is open, aes_ed AES encryption direction begins to control, after 10 times clock cycle, aes_done is effective to work, aes_dout is the results of encryption.

For synthesis of our low power AES circuit, we used the Synopsys Design Compiler. It needs 1498 clock cycles to encrypt a 128-bit data block. The required hardware complexity is estimated. From the graph, the label of digital circuit logic covers an area of 215624 units, and the combinational logic covers an area of 1656499 units, effective unit covers an area of 1872123 units.

5 CONCLUSIONS AND FUTURE RESEARCH

In this work, we proposed a compact yet high-efficient architecture for a AES cryptographic circuit and evaluated through simulation and synthesis for ASIC implementation. In order to minimize the hardware size and to optimize the throughput, the order of SubByte, ShiftRow, and AddRoundKey arithmetic functions were designed effectively and efficiently. Our architecture provides a compact and high performance AES cryptographic hardware for low power RFID system authentication. Our AES hardware has a chip size of 7829 gates. Low power techniques used are mainly based on clock and data gating. The encryption of 128 bits data requires 1498 clock cycles. The AES implementation has a chip area of has a current consumption of 15 μA at a frequency of 13.56 MHz. The results of power consumption, throughput, and functionality make our AES cryptographic hardware practical and suitable in RFID applications and other low-end embedded systems. RFID system allows the strong cryptographic authentication. With this security enhanced RFID systems, we pave the way for new security demanding applications and for the
everyday usage of RFID technology. A response authentication protocol was proposed which was integrated into the existing ISO ISO9798-2 standard.

Future work will consist in the examination of advanced authentication protocols for one-way and mutual authentication. AES circuit should be analyzed for the suitability for RFID systems and circuits can be found for this purpose.

REFERENCES


[8] Strong Authentication for RFID Systems Using the AES Algorithm Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfe


<table>
<thead>
<tr>
<th>Logic Block</th>
<th>Gate count</th>
<th>Clock cycle</th>
<th>Logic Block</th>
<th>Gate count</th>
<th>Clock cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface</td>
<td>148</td>
<td>89</td>
<td>MixColumn</td>
<td>811</td>
<td>320</td>
</tr>
<tr>
<td>S-box</td>
<td>413</td>
<td>449</td>
<td>Keyexpansion</td>
<td>257</td>
<td>50</td>
</tr>
<tr>
<td>Memory</td>
<td>2940</td>
<td></td>
<td>Controller</td>
<td>2868</td>
<td>Null</td>
</tr>
<tr>
<td>AddRoundKey ByteSub ShiftRow</td>
<td>392</td>
<td>590</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>