# SECURE ACCESS SYSTEM ON THE QorIQ NXP PLATFORM USING RSA AND DSA

Raul MALUTAN[1], Mircea POP[2], Mihaela CISLARIU[1], Monica BORDA[1]
[1]*Technical University of Cluj-Napoca, Romania*
[2]*NXP Semiconductors, Romania*
*raul.malutan@com.utcluj.ro*

**Abstract: Secure data transmission can be achieved using encryption algorithms. Combining these algorithms with authentication or communication protocols one can achieve a more secure system. Using at one of the ends a configurable platform, the system will be more scalable and the advantages will increase by adding multiple options for extension. The work proposes a secure communication ensured by RSA algorithm together with DSA between a NXP QorIQ P1025 board or an x86 machine as client and an x86 machine as server. The NTP protocol was implemented to compare the clock from the client with the clock from the server.**

***Keywords:*** *security, RSA, OpenSSL, authentication, encryption, decryption, semiconductors*

## I. INTRODUCTION

Data communication is an important aspect nowadays and considering this the protection of data from misuse is essential. Data security using public key cryptography (PKC) system was widely used in software and hardware implementations [1], [2], [3]. RSA [4] is the most adopted public key cryptography algorithm. Since it was introduced RSA has been used for establishing secure communication channels and for authenticating the identity of service providers over insecure communication mediums [5].

RSA relies on the factorization problem of mathematics that indicates that given a very large number it is quite impossible in today's aspect to find two prime numbers whose product is the given number. As one increases the number, the possibility for factoring the number decreases. The current approached topic for this paper combines perfectly the confidentiality methods ensured by the RSA algorithm through encryption and decryption functions, with the highly reliable security provided by the digital certificates and the state of the art technology available on the NXP equipment [6].

Considering this, the aim of this work was to develop a secure access embedded system. The security of the system is ensured by using the RSA algorithm, which allows the use of itself for establishing a safe communication channel but also the use for authentication with the purpose of identifying secure service providers in an unsafe network. The implementation of the algorithm on the NXP QorIQ P1025TWR platform was possible by using the OpenSSL library [7]. The system is reliable through the implementation of the NTP protocol, used for measuring the local time of the server machine. The obtained results highlighted by the OpenSSL Speed Test tool indicate the performance of the execution times for RSA algorithm by testing it with different lengths of the public key.

## II. PROPOSED SCENARIO

The system was designed as a client-server application under UNIX platform. This implies the implementation of a secure authentication system using PKC which supposed to trot out asymmetric encryption RSA using digital certificates and also SHA-1 authentication protocol on a NXP platform. The NXP QorIQ P1025E platform which is capable to support encryption operations is based on Power PC architecture and it offers good performances for networking and telecommunication applications. The power level of the applications which use QorIQ P1025 is reduced comparative with x86 platforms.

In the proposed scenario for designing the system it was used a server and two clients. The server and one of the clients are running on two x86 systems. The other client is running on QorIQ P1025E and the board has connected a keyboard on USB port. The necessary equipment used for the implementation of the system is shown in Figure 1.

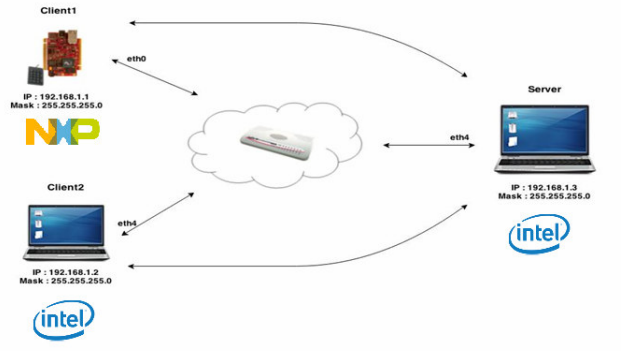


*Figure 1. A possible scenario for the secure system*

Based on this scenario a flowchart of the implementation was proposed and is shown in the following Figure 2.

Server and both clients have their certificates and their

*Manuscript received May 31, 2016; revised June 2, 2016*

private keys. When one of clients wants to authenticate in a secure manner that is (recorded), introduces a userID and a password. It forms a framework composed on userID, password, hashed with SHA-1, and system time (NTP). On this framework is applied an encryption RSA using server's public key, the next step is the transmission of the frame to the server.

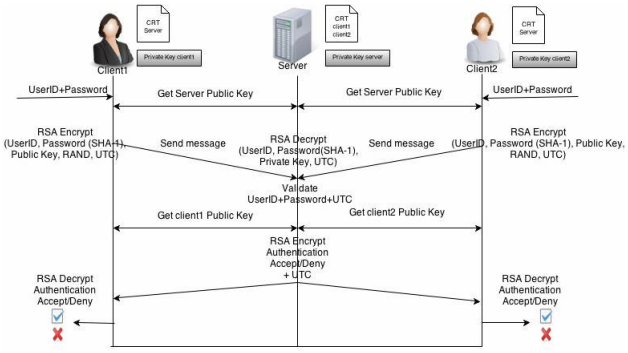


*Figure 2. System's flowchart*

The both clients are connected on different IP addresses and each client has one user and one password. So the connection of one client using an improper IP address will be fail.

After the reception of the framework, the server will decrypt it using RSA algorithm with his own private key and after that he validates the user and the password. The validation depends on the userID, password and also it depends of the local time calculated in server. In this way the both parts, client and server, are sure that the transmitted data cannot be intercepted.

The server sends an acknowledge message to the client, who decrypts the received message, specifying if the authentication is accepted or denied.

In the following paragraph the design of the system is presented.

## III. SYSTEM DESIGN

According with the flowchart from Figure 2 the first step in designing the system was to establish a secure authentication using digital certificates. For this, the following steps were implemented:

1. create and verify digital signature algorithms (DSA) using OpenSSL;
2. implement TCP Protocol on the server part.
3. create a function used for the connection of the both clients. The function is necessary in the server part because the server must listen continuous for the connection of the both clients
4. implement the certificates for client-server application. This step is about generating three certificates using OpenSSL: one for the server, one for first client, and one for the second client. Certificate management is shown in Figure 3.
5. Loading the necessary certificate for the first client on QorIQ P1025E platform. The first client needs his own private key and the server's public key which is extracted from digital certificate.
6. Implement SHA-1 algorithm using OpenSSL Library

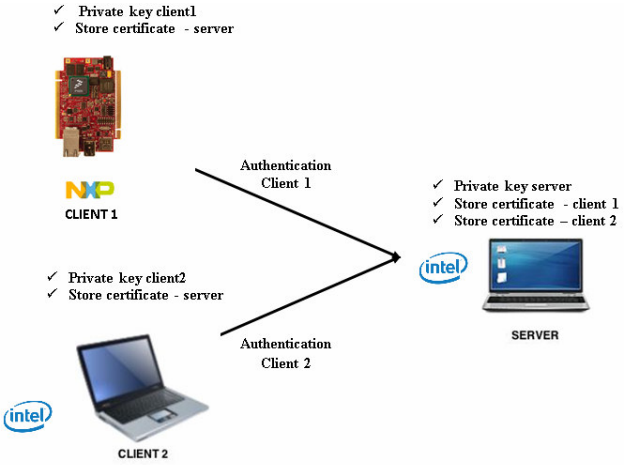


*Figure 3. Certificate management*

Once the digital certificates were generated and loaded on the NXP platform the secure data transmission using the RSA algorithm was setup and implemented according with the following steps:

1. Client implementation from the client-server architecture. This step establishes the communication between the two clients and the server using the TCP. The first client is represented by the QorIQ P1025E platform which has an input keyboard for inserting the authentication credentials, while the second client is an x86 machine.
2. USB keyboard to P1025E connection.
3. Generate prime numbers and public keys for RSA algorithm using the OpenSSL tool
5. Implement the RSA encryption and decryption for secure transmission
6. NTP (Network Time Protocol) for time synchronization. NTP is used for time synchronization between a machine with UNIX OS and another server or reference time source. In the current system for each communication point, server and clients, the access times were determined and compared, allowing an error of ±5 s. This time comparison ensures an supplementary security.
7. Load the first client on the QorIQ P1025E platform.

## IV. EXPRIMENTAL RESULTS

This section provides the experimental results for the proposed scenario. The idea was to create a successful authentication from a client with a specific username and password, which connects to the server at different time moments. This way one can observe the variety of encryption randomization.

Also in this section will be presented performance testsof the RSA public key algorithm and application for generating prime numbers, on x86 platform and the NXP QorIQ P1025E platform. The NXP board uses a single core CPU with 533,33 MHz, 512MB DDR3 memory, while the x86 machine has an Inter Core 2 Duo processor at 2,2GHz.

The first step in the proposed scenario consists of clients authentication on the server from the two different platforms with the established credentials and also with the time synchronization. The margin of error allowed is ±5 s and it

tries to discover possible attacks which may appear during the transmission. A successful authentication and exchange of messages can be observed in Figure 4.

Once the server does the data validation it sends to the client on its IP address a confirmation message with the serves local time and a successful authentication encrypted message. The user decrypts the message in order to read the authentication status and validates its time with the server's time. The analysis of the exchanged messages between server and client was done with the Wireshark tool.

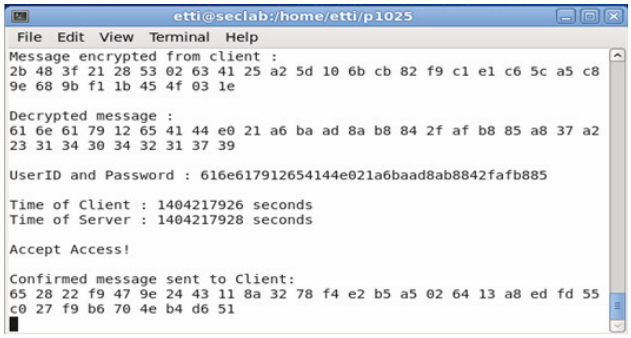


*Figure 4. Client authentication*

The Digital Signature Algorithm was tested using the web server NGINX on both plaforms, x86 and NXP, and an Apache Benchmark client. The analysis consists of measuring 5000 connections with different lengths for the private key. For all the connections the following parameters were measured: *time for tests*, *requests per second* and *time for request*. Figure 5 shows the results obtained on the x86 machine and QorIQ P1025 board for the number of operations per Watts. So, from the power consumption point of view, the x86 machine is better, but only because it uses a dual core processor.

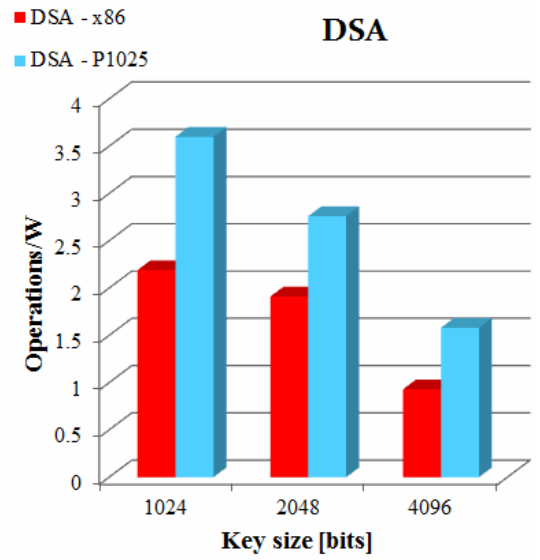


*Figure 5. Comparison of the number of operation/s vs key size for the DSA implementation on x86 and P1025*

As for the scenario part where the RSA encryption and decryption was designed and implemented the test were done for different key lengths: 1024, 2048 and 4096 bits, and also considering the size of the data frame: 16, 64, 128, 256, 512, 936, 1024, 1960, 2048 and 4008 bits/frame.

Figure 6 (a) and (b) presents the obtained results for the execution time of encryption and decryption when 1000 times the data was sent from the client to server and reverse from each platform. The computation speed of the 2GHz x86 platform is obvious from the results obtained, while the P1025 NXP platform ensures a better scalability.

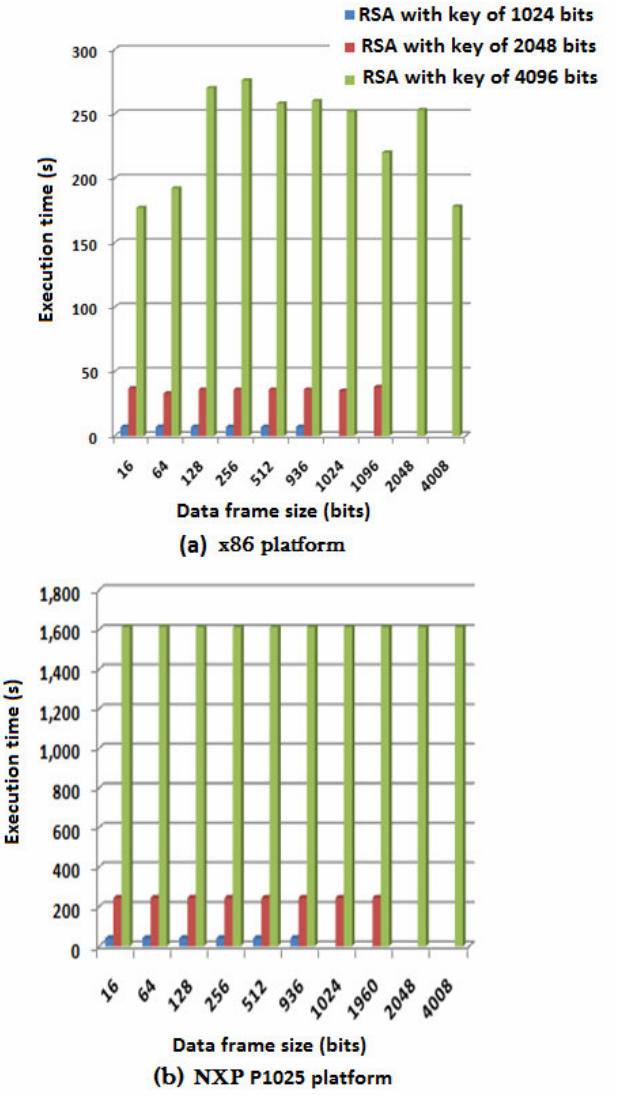


*Figure 6. Results of execution time for RSA encryption and decryption on both platforms for 1000 iterations*

Prime number generating was done according with the *bignum* [8] toolkit from the OpenSSL environment. This tool offers the possibility to compute the execution time for x86 platform and P1025 board.

For the prime number generator in this scenario was considered the size of the prime number in bits and the number of executions in seconds.

For the x86 machine one can notice from Fig. 7 a rapid increase correlated with the size of the prime number.

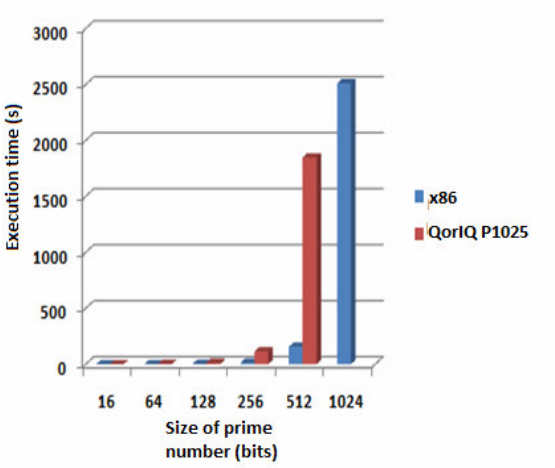


*Figure 7. Results of execution time for prime number generation in 100 iterations*

For the QorIQ P1025 board the results were stopped after the 512 bits size prime number due to the large difference between the execution time obtained on the x86 platform and NXP platform for 1024 bits.

## V. CONCLUSIONS

The proposed system provides a secure communication between a QorIQ P1025TWR board which has the role of the client and an x86 machine as the server. Another scenario has the x86 machine as client and another machine as server. In both cases the computed execution time and number of operations per Watts, meant to determine the power consumption of the algorithm, gave better results for the use of a client on x86 platform. This was due to the hardware specification of this machine 2 GHz dual core versus 500 MHz single core processor. The security of the system was dual ensured by the encryption algorithm RSA and an implementation of the NTP protocol which was set to compare the clock from the client with the clock from the server, and to give access to the client only if the time difference between these has an error margin of 5 seconds. On the authentication step it was used a digital signature algorithm with the help of OpenSSL tool, and the system was analyzed for different key lengths in order to achieve the best configuration.

## REFERENCES

[1] A. Pellegrini, V. Bertacco, T. Austin, "Fault-based attack of RSA authentication", *Proceedings of the Conference on Design, Automation and Test in Europe,* pp. 855-860, 2010

[2] Qasem Abu Al-Haija, Mahmoud Smadi, Monther Al-Ja'fari, Abdullah Al-Shua'ibi, "Efficient FPGA Implementation of RSA Coprocessor Using Scalable Modules", Proceedings of the 9th International Conference on Future Networks and Communications (FNC'14), Volume 34, pp. 647–654, 2014

[3] H. Seo *et al.* "Montgomery Modular Multiplication on ARM-NEON Revisited", *Information Security and Cryptology-ICISC,* pp. 328-342, 2014

[4] V. Mainanwal, M. Gupta, S.K. Upadhayay, "Zero Knowledge Protocol with RSA Cryptography Algorithm for Authentication in Web Browser Login System (Z-RSA)", *IEEE Fifth International Conference on Communication Systems and Network Technologies (CSNT),* pp. 776-780, 2015

[5] R. Amin, G.P. Biswas, "An improved RSA based user authentication and session key agreement protocol usable in TMIS", *Journal of Medical Systems*, vol 39(8), pp. 1-14, 2015

[6] P1025: QorIQ P1025/16 Single- and Dual-Core Multi-Protocol Communications Processors http://www.nxp.com/

[7] A. Shackelford, *Securing the Application*, Springer, 2015

[8] Bignum library in OpenSSL

https://www.openssl.org/docs/manmaster/crypto/bn.html