Optimization of elliptic curve digital signature algorithm (ecdsa) and its implem...

Technology, Innovation

Optimization of Elliptic Curve Digital Signature Algorithm (ECDSA) and Its Implementation

Abstract: In the present era, developing high-level application programs has been a major concern for the programmers. Elliptic curve digital signature algorithm (ECDSA) is one of the fastest growing fields in cryptography. On the internet communication, securing e-commerce and other online transactions requiring authentication is a necessity. Optimization involves making a program bug-free, reduced time and space complexity etc. This paper presents an optimization of the ECDSA algorithm using C code optimization techniques “ Loop Unrolling” through which the execution speed of the process is improved. Un-Optimized and Optimized ECDSA is implemented using Xilinx ISE Design Suite 14. 5 using Virtex6 and the results are obtained. The execution time for hardware implementation of optimized ECDSA code is improved by. Keywords: C code optimization techniques, ECC, ECDSA.

In daily life everything happens over the internet, we send email, online chat, purchase good and products over the e-commerce website their security is most important. Information security has the greatest importance in a world in which communication over open networks and storage of data in digital form play a key role. Radio Frequency Identification (RFID) tags, sensors, mobile phones, appliances, etc., are provided with special identifiers and the capacity to communicate with each other over a network to reach common objectives without requiring human interaction. The importance of information security has grown because new technologies have made

accessing and misusing confidential information easier and more profitable. We want to keep certain things like our internet passwords, credit card numbers, banking information and business documents from getting into the wrong hands. Personal information is important to us but not for criminal, they misuse our information and get profit. If our important file, work documents, photos, customer details go into the wrong hand’s embracement and inconvenienced etc. happened. To protect the increasing criminal on networks cryptography is required.

Cryptography deals with making communications secure. It is the art and science of making a cryptosystem that is capable of giving information security. Cryptography deals with the real securing of digital information. It refers to the plan of components based on scientific calculations that give essential information security services. Cryptography permits individuals to keep confidence in the electronic world. It is used in many applications like banking transactions cards, computer passwords, and e-commerce transactions.

The advantages of cryptography are Confidentiality: Ensuring that no one can read the message except the intended receiver.

Integrity: Information cannot be altered.

Non-repudiation: Sender cannot deny his/her intentions in the transmission of the information at a later stage.

Authentication: The process of proving the sender and receiver identity. There are 2 types of cryptography symmetric key cryptography and asymmetric key cryptography.

Symmetric-key Cryptography: Both the sender and receiver share a single key.

The sender uses this key to encrypt plaintext and send the ciphertext to the receiver. Blowfish, AES, RC4, DES, RC5, and RC6 are examples of symmetric encryption. The most widely used symmetric algorithm is AES-128, AES-192, and AES-256 these are the Symmetric-key Algorithms.

Asymmetric-key Cryptography: In Public-Key Cryptography two related keys (public and private key) is used. Public key freely distributed, while private key, remains a secret. Asymmetric encryption is mostly used in day-to-day communication channels, especially over the Internet. RSA, DSA, Elliptic curve techniques are the popular asymmetric-key algorithms. Due to the better performance of ECC, the Elliptic curve has been adapted for several cryptographic schemes, such as Key agreement scheme: ECDH, Encryption scheme: ECIES, Digital signature scheme: ECDSA. In this paper, we are working on the ECDSA algorithm. The ECDSA offered remarkable advantages over other cryptographic system mentioned by [4].

• It provides greater security with smaller key sizes.

• It provides effective and compact implementations for cryptographic operations requiring smaller chips.

• Due to smaller chips less heat generation and less power consumption.

• It is most suitable for machines having low bandwidth, low computing power, less memory.

• It has easier hardware implementations.

ECC algorithm

With the development of E-commerce and E-government, the demand for a fast and secure public key cryptography algorithm was growing increasingly. Elliptical curve cryptography (ECC) could be a public key encryption method based on elliptic curve theory that can be utilized to make faster, smaller, and more effective cryptographic keys. ECC produces keys through the properties of the elliptic curve equation rather than the traditional strategy of generation as the product of exceptionally huge prime numbers. A 160-bit key in ECC has the same security level as 1024-bit key in RSA. ECC has some additional advantages such as a more compact structure, a lower bandwidth, and faster computation that all make ECC usable in both high-speed and low-resource applications [1]. It is widely used for mobile applications. ECC was developed for a mobile e-business security provider, a manufacturer of integrated circuitry and network security products.

ECDSA algorithm

In this paper, we have to use the digital signature algorithm based on an elliptic curve. Elliptic Curve Cryptography Elliptic Curve Digital Signature Algorithm (ECDSA) is a variant of Digital Signature Algorithm (DSA) which

uses Elliptic Curve techniques (EC). Surely, both algorithms (ECDSA and DSA) are relatively the same. Digital Signature Algorithm (ECDSA) provides high security. Faster computation and lesser processing power, storage space and bandwidth are accomplished by ECDSA because of its smaller key size. Elliptic curve digital signatures are commonly used for software distribution, financial transactions, and vehicles, mobile and in other cases where it is important to detect forgery or tampering. Elliptic curve digital signature algorithm comprises of three calculations:

1) ECDSA Key Pair Generation Fig. 2. Key Pair Generation This step needs one pseudo-random number generator to choose d and one point multiplication to compute Q for more security (Fig. 2). The sender does the following: 1. Select an elliptic curve E defined over Fp. The number of points in E should be divisible by a large prime n. 2. Select a point generator P € E (a, b) of order n. 3. Select a statistically unique and unpredictable integer d from [1, n – 1]. 4. Compute the point Q= d. P 5. Sender’s public key is Q; sender’s private key is d.

2) ECDSA Signature generation This step needs a pseudo-random number generator to choose k, one point multiplication, one modular reduction, one hash function, addition and modular division (Fig 3). Fig. 3. Signature generation The signature is the set (r, s). To sign a message m, the sender does the following: 1. Select a statistically unique and unpredictable integer k from [1, n – 1]. 2. Compute k. P = (x1, y1). 3. Compute r = x1 mod n 4. Compute e = h (m) with e the message digest and h the hash function 5. Compute s = k-l. (e + d. r) mod n

3) ECDSA Signature verification Fig. 4. Signature verification [3] To verify sender’s signature (r, s) on m, the receiver must compare v and r (Fig. 4). If v = r then the signature is valid else It is invalid. It should do the following: 1. Compute e = h (m) 2. Compute ul = e. s-1 mod n 3. Compute u2 = r. s-1 mod n 4. Compute ul. P + u2. Q = (xl, y2) 5. Compute v = x2 mod n 6. Accept the signature if and only if v = r.

Software implementation of un-optimized ECDSA algorithm

The ECDSA algorithm written in C language, The Dev- C++ 5. 11 compiler used for the software implementation of ECDSA algorithm. The obtained hash is given to user 2, where the user 2 verify the signature using the public key of user 1 and the private key of user 2. The obtained signature is matched with the received signature, and both are found to be same thus the signature is verified. When by using the same public and private keys of the same user the signature not match, thus it indicates that the algorithm is working properly.

Software implementation of optimized ECDSA algorithm

During the design phase of the software application, the use of created programs has dependably been the area of thought. In general, a computer program may be optimized so that it executes more rapidly. To enhance the performance of the large complex applications, several code optimization techniques are being provided in C, those techniques like Loop unrolling, Strength Reduction, Code Motion, Constant folding, Constant propagation, Dead Code Elimination, Common Sub-Expression Elimination have been

studied. Optimization using these techniques occurs that advantages like Execution Faster, Efficient memory usage, Yield better performance. Loop unrolling, constant folding these are two techniques applicable in C code of ECDSA algorithm so that using these two techniques optimize ECDSA algorithm and increase the speed. A. Optimization using loop unrolling Techniques Loop unrolling, also known as loop unwinding, is a loop transformation technique that attempts to optimize a program's execution speed. The advantages of loop unrolling are its increments program efficiency and Reduce loop overhead. For example, int main(void) { for (int i= 0; i <5; i++) printf(" Hellon"); //print hello 5 times return 0; } The above for loop printed the Hello word by executing 5 times this is the time-consuming task, using the loop unrolling techniques this execute rapidly shown in below int main(void) { // unrolled the for loop in program 1 printf(" Hellon"); printf(" Hellon"); printf(" Hellon"); printf(" Hellon"); printf(" Hellon"); return 0; } By using loop unrolling techniques the five for loop unrolled of ECDSA algorithm so that code executed rapidly.

Hardware implementation

Both software implemented C code for ECDSA algorithm is converted into Verilog using the Vivado high-level synthesis 14. 4 then implemented on Xilinx ISE design suite 14. 5. A. Conversion of Un-Optimized ECDSA Algorithm from C to Verilog Creating a new project on Vivado HLS, Add/remove C-based sources files (design specification). Create Vivado HLS solution for selected technology. Open the source from the left side and write Un-optimize ECDSA c code after that save this file. Then go to the solution from the upper side

tab and select run all active solution. Take the generated code from the folder solution à Synthesisà Verilog. Get the RTL code from the solution folder and run it on Xilinx ISE design suite 14. 5. Create Vivado HLS solution for selected technology. Open the source from the left side and write Un-optimize ECDSA c code after that save this file. Then go to the solution from the upper side tab and select run all active solution. Take the generated code from the folder solution à Synthesisà Verilog. Get the RTL code from the solution folder and run it on Xilinx ISE design suite 14. 5.

It has been analyzed that after executing the normal ECDSA C code and optimized ECDSA C code on Dev C++ software, as well as on the hardware the execution time has been decreased. The execution time for hardware implementation optimized ECDSA algorithm is improved by 73. 44% thus overall execution time is reduced for optimized ECDSA.