Embedded Cryptography

Jia Chen, Hongyi Zhen, Surya Teja Sharma, Ashley Lee, Hung Huynh

Abstract-- The Internet of Things (IoT) is expected to connect billions of devices to become the future of technology. The increased number of data and communication nodes is expected to generate mountains of data. The security of data can be extremely vulnerable. The IoT devices are essentially embedded computers smaller in size and low powered. Conventional encryption algorithms, such as RSA, are generally computationally expensive due to their complexity and extensive key size, essentially surpassing the energy capacity of IoT devices. Less complex algorithm, on the other hand, may compromise security. This paper introduces the implementation of Elliptic Curve Cryptography (ECC). Security using ECC uses smaller key sizes is efficient for both private and public operations. IoT devices with low computational power makes ECC advantageous when you need to embed security.

Keywords-- IoT; Security; Encryption; Elliptic Curve Cryptography;

1. Introduction

Next generation computing platforms are embedded within physical devices and people, forming a network called Internet of Things (IoT). These embedded systems increased in complexity and network connectivity, creating a much larger attack surface and new points of entry for malicious attackers [1]. As Internet of Things devices are optimized for lower power consumption and affordability, many have less than optimal computing resources, energy consumption, and memory to support proper network security [2]. Millions of devices, with little to no protection, share the same hard-coded cryptographic keys and certificates. This exposes them to various types of malicious attacks. Information security then becomes as a significant concern. If one device is hijacked remotely, an attacker can possibly access hundreds of thousands of other interconnected devices - including ones from different manufacturers. The attacker would then be able to decrypt network traffic to extract usernames, passwords, and other sensitive data. An embedded device would require security protocols and mechanisms for transferring data throughout a network and should also enforce security to resist unauthorized access of sensitive data from the device.

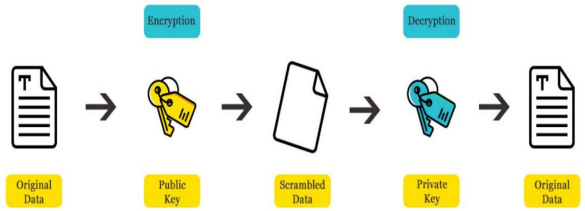

Figure 1. Public key cryptography (https://ieeexplore.ieee.org/document/7993462)

Embedded cryptography uses authentication and encryption protocols, relying on public key infrastructure to provide secure communications (Figure 1). Proper encryption/decryption would allow the sender to encode information using its public key and the receiving party uses its private key to decrypt this information. Attackers would not be able to decipher

encrypted data. Authentication allows intended receivers to identify the file origin by validating the identity of the sender.

The RSA algorithm is one of the most popular public key cryptography system. Its security mechanism relies on mathematical proofs that multiplication is fast and factoring is slow. With specialized algorithms getting more efficient at cracking prime factorization, it becomes necessary to increase the key size in RSA to achieve security [3]. Mobile phones and embedded systems lack the computation power to support ever-growing RSA keys. With smaller key sizes while keeping the same level of security, ECC is able support less memory usage, faster key generation, faster certificate processing, etc.

In this paper, we will discuss ECC as an efficient algorithm to improve the security with less memory and lower computational cost, which is better than cryptosystems that rely on more computationally expensive operations.

2. Background

Public-key cryptography, also known as asymmetric cryptography, (Figure 1) is an encryption scheme that uses two non-identical keys that are mathematically related: a public key and a private key [4]. The public key is used to encrypt the entity whereas the private key is used to decrypt it. ECC is one of many types of public key cryptography that is based on the discrete logarithm problem called Elliptic Curve Discrete Logarithms (ECDL).

$$y^2 = x^3 + ax + b$$

Figure 2: Elliptic Curve Equation

ECC constructs a finite field out of the set of solutions to an elliptic curve equation (Figure 2) with an identity element that corresponds to the point at infinity [5].

There are two types of finite fields that the curve will be taken over: prime and binary. In prime fields, operations are computed modulo prime number p whereas operations are computed modulo an irreducible polynomial in binary fields [6]. Since binary fields use irreducible polynomial, operations under binary fields tend to be faster than ones under prime fields [6]. Therefore, prime fields are most commonly used.

ECDL is a type of trapdoor or one-way function. The function is relatively easy to go in one direction, but it is infeasible to reverse the computation [7]. ECC's trapdoor function makes it almost impossible for user to figure out the private key with just the public key [5]. Therefore, it is simple to calculate the last point of the curve if other two points are known. However, if only one point is known, it is infeasible to deduce the other two points. A simple example of a trapdoor function is factorization. Given two prime numbers p and q, it is easy to compute their product, n. However, given only the value of n, it is very difficult to deduce the values of p and q (Figure 3).

$$f_p(q) = p * q \text{ where } p = 6203 \text{ and } q = 9467$$
$$f_p(q) = 6203 * 9467$$
$$f_p(q) = 58{,}723{,}801$$

Given only the value of $f_p(q)$, it is difficult to deduce the values of p and q

However, given $f_p(q)$ and p, it is easy to compute the value of q

$$q = \frac{f_p(q)}{p} = \frac{58{,}723{,}801}{6203} = 9467$$

Figure 3: Example of Trapdoor Function

The elliptic curve is a symmetric curve about the x-axis (Figure 4). According to the property of elliptic curve, if a straight line is drawn through the curve, the line will only intersect the curve at three points (A, B, and C). When A and B are known, it is easy to compute C. However, when only C is known, it is nearly impossible to compute A and B.

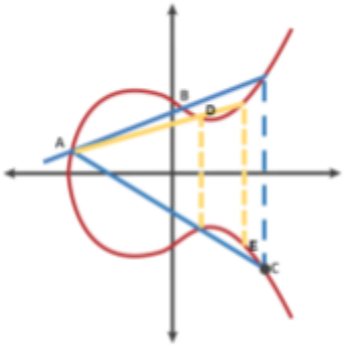

Figure 4: Elliptic Curve (from Wagner, 2018)

In the algorithm, the dot function is repeatedly utilized to yield a new point on the curve to jump to until the endpoint is found [7]. Starting at point A, the dot function is used to find -C. Under the group law on elliptic curve, the inverse of the new point, denoted by the dotted line, leads to C. Then, the dot function between A and C is calculated to yield the third point -D and the inverse of -D leads us to D. Likewise, this computation is repeated n number of times until the end point E is found.

The public key is the starting point A and ending point E and the private key is the number of hops taken on the curve to end up from starting point to ending point. When the user is only aware of the starting and ending points, it is extremely difficult to deduce the number of hops due to inherent characteristics of the trapdoor function. Therefore, encryption is easy to compute but decryption is very difficult, near impossible, to compute without the private key [6]. This forms the basis for the trapdoor function of ECC.

3. Threat Model

A few of kinds of attacks that are mentioned in the attack tree(Figure 6) can be mitigated by good policy making and usage of the ECC to ensure better security.

Potential Threat	**Risk/Impact**	**Prevention/ Mitigation**
Obtaining Private Key through Brute Force	Unauthorized access and control	- Rotation of keys should be enforced as a policy
Forging Certificate of trusted device	Malicious device part of trusted network	- Certificates should be issued by trusted 3rd parties and a central list of device-certificate mapping should exist

The system can be open to Distributed Denial of Service attack	System will not allow legitimate user to access resources	- Root login should be disabled - IPs should be blocked by utilizing Tools/Rules
Extracting Keys from Firmware	Unauthorized access and control	- As the generation of keys in no longer computing intensive, periodic changes to keys will render this threat to be ineffective

Attack Tree for the Embedded Cryptography

The following diagram is the attack tree which outlines the potential threats and attacks to the embedded cryptography.

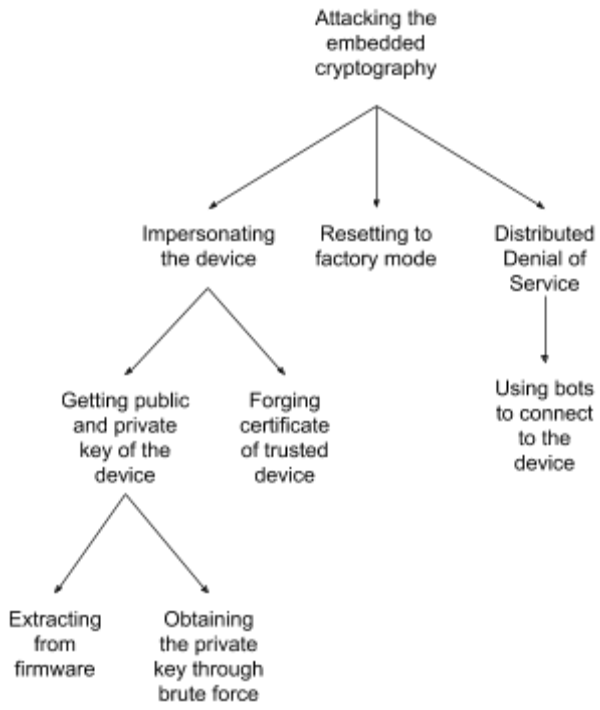

Figure 6: Attack tree

4. Design

The system proposed will abide by the following security principles:

1. Open Design
2. Complete Mediation
3. Fail Safe defaults
4. Economy of mechanism

In the system, the components involved are:

1. Device - Sensors/ Actuators
2. Gateway- Routers/ Edge Computing Devices
3. Cloud - Store for collected data
4. End User applications to access Devices

5. Implementation

Implementing Diffie Hellman Elliptic curve for key exchange requires sharing of the domain parameters required for establishing communication protocol. Now, according to the Public Key Exchange infrastructure, following steps are followed-

1) **Exchange of domain information** : These parameters include the curve equation, prime number (used to make the curve prime and have a bound), Generator point, order of generator point and the cofactor.
2) **Generation of private keys** : Alice and Bob need to generate their private keys, in this case, they pick a number from range 1 to n.
3) **Compute points on curve**: With the respective private keys, Alice computes a point A and Bob computes a point B. They exchange with each other these points as public information.

4) **Agreeing on a shared secret(mutual point)** : Alice multiplies her private key to point B and Bob multiplies his private key to A. Now both parties have the same point which can be used for communication.

In the case of securing IoT devices, this exchange typically happens between devices and gateways, gateways and cloud, cloud and user, where the devices collect data and send it back to their master.

6. Evaluation

After reviewing the implementation of ECC, we can easily tell that ECC demonstrates significant improvement compared to the RSA algorithm.

Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Table 1: NIST Recommended Key Sizes

https://www.globalsign.com/en/blog/elliptic-curve-cryptography/

Firstly, one main advantage of ECC algorithm is that we can use smaller key for the same level of security compared to RSA algorithm. In general, the security of a 160-bit key of ECC is equivalent a 1024-bit key of RSA, and 256 bits is equivalent to 3072 bits. So it has an advantage in embedded systems like IoT which have limited resources. In addition, we generate key faster in ECC algorithm. It's just randomization and point multiplication. But in RSA, it is much more complex including the calculation of totient and satisfaction of congruence relation.

Therefore, using ECC in embedded system will efficiently improve the security with less memory and lower computational cost.

7. Related Solutions

ElGamal encryption is another type of public-key cryptography based on the discrete logarithm problem. This scheme uses the discrete-logarithm assumption where it is hard in some groups to find x given g^x mod n. The security of the algorithm relies on the level of difficulty of computing discrete logs in a large prime modulus [8]. Therefore, ElGamal encryption is slower than RSA, but ElGamal decryption is faster than RSA [9].

A new design for the ECC cryptosystem using the H-AES-LCG generator function can improve the domain parameters in random, efficient, and secure manner. ECC is vulnerable to attacks by exploiting its public parameters so these parameters should be selected safely to guard against all attacks. The proposed methodology is faster and provides many positive aspects such as enhancements in the key exchange compared with Diffie-Hellman key exchange and ECC performance [10].

8. Conclusion

In this paper, we discussed about a solution to improve embedded cryptography using ECC, which will significantly reduce the computational cost of the key generation without compromising the security strength. The solution will be helpful when implemented in an IoT system where the

computational resource of the router and end devices are limited. Having said that, due to the lack of wide usage compared to RSA, this newer algorithm could theoretically have unknown weaknesses and have lesser acceptance in production systems.

References

[1] Mehran Mozaffari Kermani, Meng Zhang, Anand Raghunathan, and Niraj K. Jha. "Emerging Frontiers in Embedded Security." 2013 12th International Conference on Embedded Systems, Pune, India, (January 2013). https://ieeexplore.ieee.org/document/6472640

[2] Konstantinos Fysarakis, George Hatzivasilis, Konstantinos Rantos, Alexandros Papanikolaou, and Harry Manifavas. "Embedded Systems Security Challenges." Measurable security for Embedded Computing and Communication Systems, Lisbon, Portugal, (January 2014). https://www.researchgate.net/publication/259619970_Embedded_Systems_Security_Challenges

[3] Sullivan, Nick. "A (relatively easy to understand) primer on elliptic curve cryptography." arstechnica (October 2013). https://arstechnica.com/information-technology/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/2/

[4] Wagnon, John. "Real Cryptography Has Curves: Making The Case For ECC." F5 DevCentral, (July 2016). https://devcentral.f5.com/articles/real-cryptography-has-curves-making-the-case-for-ecc-20832.

[5] Roeder, Tom. "Asymmetric-Key Cryptography." (September 2007). https://www.cs.cornell.edu/courses/cs5430/2013sp/TL04.asymmetric.html

[6] Dana Neustadter, Tom St Denis. "Elliptic Curves over Prime and Binary Fields in Cryptography." (2008). https://www.fields.utoronto.ca/programs/scientific/07-08/cryptography/dana_neustadter.pdf

[7] Wagner, Lane. "(Very) Basic Elliptic Curve Cryptography." Good Audience. (June 2018). https://blog.goodaudience.com/very-basic-elliptic-curve-cryptography-16c4f6c349ed?fbclid=IwAR3TREx_Kt_sBoyTX8-QHcYLY2RN12fqfAEKZtYD0f9rPLuwXf9OS_b625

[8] Zengqiang Wu, Di Su, Gang Ding. "ElGamal algorithm for encryption of data transmission" IEEE (September 2015). https://ieeexplore.ieee.org/document/7231798

[9] Steven Li. "A Whirlwind Tour of Modern Cryptography." (May 2017) https://medium.com/@User3141592/notes-on-computational-cryptography-98db5f2908f1

[10] Kawther Esaa Abdullah, Nada Hussein M. Ali. "Security Improvement in Elliptic Curve Cryptography." (2018) http://thesai.org/Downloads/Volume9No5/Paper_16-Security_Improvement_in_Elliptic_Curve_Cryptography.pdf