Elliptic curve cryptography in cloud computing security

Manu Gopinathan (manugopi92@gmail.com)
Øyvind Nygard (oyvind2302@gmail.com)
Kjetil Aune(aune.kjetil@gmail.com)

December 1st, 2015

1 Abstract

Cloud computing is a technological advancement that has been growing swiftly during the last decade. In simple terms, cloud computing is a technology that enables shared, remote, on-demand and ubiquitous access to services through the Internet. It enables consumers to access applications and services that reside on remote servers, without having to allocate large amounts of storage space on their own computer and without the need for extensive compatibility configurations. Many such cloud applications provide services that are meant to handle sensitive user data and thus the protection of this data in terms of access and integrity is of major concern. Space- and time complexity of encryption algorithms can prove to be imperative when it comes to system performance. In this paper we will briefly present how elliptic curve cryptography (EEC) works, and then describe the advantages of it and how it can be used as an encryption solution to security related issues in cloud computing.

2 Introduction

In this section we will briefly describe the notion of cloud computing to aid us in the discussion of ECC in cloud computing later. According to the National Institute of Standards and Technology (NIST), essential characteristics for a service based on the cloud computing model are [1]:

1. On-demand self-service: The consumer can provision service capabilities, such as server time and network storage, without actively interacting with the service provider.
2. Broad network access: the service is available through common client platforms that have network access, like mobile phones, laptops etc...
3. Resource pooling: Resources serve multiple consumers and dynamically reassigns, according to the demands of the consumer. Examples of resources are storage and network bandwidth.
4. Rapid elasticity: This term is related to the scalability of the service. On the consumer side, the cloud computing resources may seem infinite, even though allocation and de-allocation of resources may occur at the provider's end. In other words, service capabilities are provisioned and released to accommodate increase or decrease in demand.
5. Measured service: The provider of the cloud service can monitor and control the amount of resources used by, or available to, the consumers. Thus different options can be made available to the consumers, depending on the their different needs.

Furthermore, cloud computing services can usually be placed in one of three different categories of service models:

1. Software as a service: The consumer can use applications running on the provider's cloud servers. The underlying cloud infrastructure aspects of the service, such as network, storage and operating systems, are not controlled by the consumer. A possible exception is user-specific application configuration settings. Examples are: Dropbox and Google Docs.

2. Platform as a service: The consumer can create applications using programming languages, libraries and tools supported by the provider and deploy these applications onto the provider's cloud platform. As before, most of the underlying cloud infrastructure is still not controlled by the consumer. However, the consumer has control over deployed applications. Examples are: Microsoft Azure and Google App Engine.

3. Infrastructure as a service: The consumer is provided with resources like processing power, storage and networks to be able to deploy and run applications and operating systems. The consumer has control over operating systems, storage and deployed applications. Examples are: Amazon EC2 and Rackspace

As with most applications, cloud computing services face many security challenges. The ones we would like to address in this paper are the those related to integrity, confidentiality and authentication, all of which concern data protection.

3 Elliptic Curve Cryptography

3.1 The basics

Elliptic curve cryptography (ECC) is a cryptographic scheme that uses the properties of elliptic curves to generate cryptographic algorithms. In the 1980s Koblitz and Miller proposed using the group points on an elliptic curve defined over a finite field in discrete logarithmic cryptosystems.

An elliptic curve is the solution set over a non-singular cubic polynomial equation with two unknowns over a field F. In short terms it is a discretized set of solutions to a curve that is in the form:

$$y^2 = x^3 + ax + b \tag{1}$$

These curves holds the property that if you draw a straight line that intersects the curve in two points, it will also intersect the curve in a third point that is either on the curve or the point of infinity (also referred to as the neutral element) (figure 1).

Another important property of elliptic curves is that they are symmetric over the x-axis. That means that if you have a point P(x, y) then -P will be (x, -y) (figure 2).

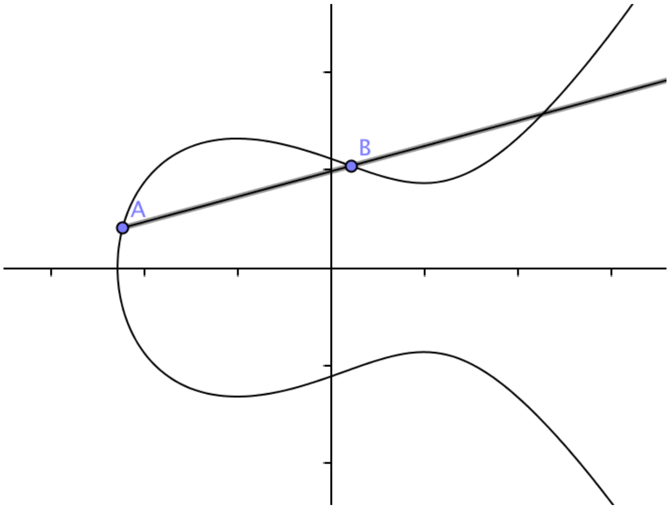

Figure 1: A straight line through 2 points will always intersect the elliptic curve in a third point.

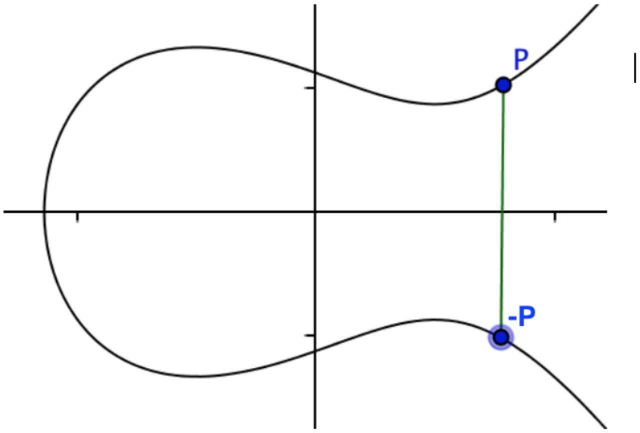

Figure 2: Symmetry over the x-axis.

Using these properties one can define some interesting and useful arithmetic rules. We will now briefly explain how point addition over elliptic curves is done, as this is used for key generation.

Suppose that you have a point A and a point B on an elliptic curve and you want to perform an addition of these two points. Then you draw a line from A through B. This line will intersect the curve in a third point. Take this third point and mirror it over the x-axis and that will be the result of the addition.

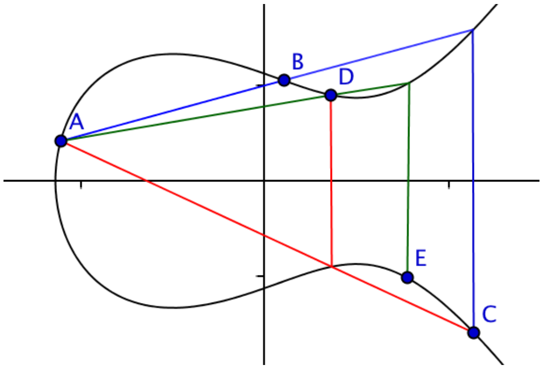

Figure 3: Elliptic curve point addition.

Further, let's assume that the point B on the figure is the result of point A added to itself. Then we have the following properties (as shown in figure 3):

$$A \oplus A = B \tag{2}$$

$$A \oplus B = A \oplus A \oplus A = C \tag{3}$$

$$A \oplus C = A \oplus A \oplus A \oplus A = D \tag{4}$$

$$A \oplus D = A \oplus A \oplus A \oplus A \oplus A = E \tag{5}$$

Elliptic curve cryptography is defined using the following domain [6]:

$$D = (q, Fr, S, a, b, P, n, h) \tag{6}$$

Here q is a prime number or a power of 2 to the k, Fr is the field representation, S (optional) is a bitstring of at least 160 bits [8], a and b are the curve coefficients, P is the base point (Px, Py), n is the order of P and h is the co-factor co-efficient [8]. Key generation and validation works as follows[6]:

Algorithm 1 Key pair generation

```
Select $d \in [1, n-1]$.
Compute $Q = dP$.
Return ($Q$, $d$), $Q$ is the public key and $d$ is the private key.
```

Algorithm 2 Public key validation

```
Verify that $Q \neq d$.
Verify that $Qx$ and $Qy$ are properly represented elements of $Fq$.
Verify that $Q$ satisfies the elliptic curve equation defined by $a$ and $b$.
Verify that $nQ = \infty$.
If any verification fails then return("Invalid"); else return("Valid").
```

3.2 Advantages of elliptic curve cryptography

Currently, there exist many different cryptosystems that can be used to secure a system. RSA is one of the most widely used cryptosystems to date. Surely, this indicates that RSA provides a sufficient level of security? So why use ECC?

In 2012 Mark Knight, from Thales e-Security, stated that:

"(...) cryptography can be considered an arms race, where potential attackers have access to increasingly powerful computing resources and techniques to try and calculate key values. The result is that some algorithms and key sizes that were considered secure in 2010 are unlikely to be secure from attack in 2030."[3]

This is not an unreasonable thought and if this is the case, a reasonable and simple solution could be to use a longer key. In general, it does make sense to assume that larger keys provide greater resistance against attacks. However, this can lead to some undesirable effects. In the case of the RSA, doubling the length of the key reduces the performance by a factor of 5-7 [3]. The advantage of ECC becomes clear when observing the security level that keys of different bit sizes provide.

ECC key size	RSA key size	Key size ratio
160	1024	1:6
224	2048	1:9
256	3072	1:12
384	7680	1:20
512	15360	1:30

Table 1: Comparison of key size, in bits, for RSA and ECC [8, 9]

Table 1 clearly shows that there is an increase in performance of ECC compared to RSA. Not only does smaller ECC based keys produce stronger level of security, but the ratios show that if we double the ECC based key size, the RSA key size has to be increased to more than the double. We also see that this escalates for even greater key sizes. Mark Knight also states that because of this, key generation can be a 1,000 times faster with ECC than with RSA. A combination of these benefits results in a reduction of network, memory and storage overheads.

4 Application of ECC in cloud architecture

When it comes to the application of ECC to cloud computing security, Tirthani and Ganesan propose a simple architecture based on the Diffie-Hellman Key Exchange and ECC to secure a cloud system [2]. In this section we present their proposed architecture for a client/cloud server system, which uses a four step procedure that consists of connection establishment, account creation, authentication and data exchange.

Establishing the connection to the system and the creation of an account for a first-time user constitute the two first steps. When a first-time user accesses the system for the first time, an initial connection can be made using HTTPS and SSL protocols. During account creation, the cloud server will generate a unique user ID and the private/public key pair necessary for ECC encryption.

The third step is the authentication of a user that logs in to the system. This is done by having the user provide the user ID that was created during account creation. The final step, the data exchange, is done by using the Diffie-Hellman key exchange protocol. When the client wants to retrieve data from the server, the client's query is stored in a file which is encrypted with the servers public key and the client's private key. The encrypted file is sent to the server, where the file is decrypted using the server's private key to retrieve and process the query. Thereafter, the server encrypts the resulting data with the server's private key and the client's public key, and sends this back to the client who can retrieve the queried data by decrypting the package with his/her own private key.

5 Conclusion

Cloud computing security is of major relevance as a result of the growing number of services that have emerged in the recent years. Most of these services have to handle and protect the consumers' sensitive data. From the points presented, it is evident that as intruders gain more resources and develop new techniques to crack our systems, we have to modify them to be more robust against these attacks. Cryptosystems like the RSA, which is widely used today, seem to be reaching a point where they are outdated when performance is on the line. Elliptic curve cryptography has been around for a while, but has not garnered widespread usage yet, compared to other cryptosystems. However, ECC has considerable advantages when it comes to security performance and is quite compatible with predominant cryptographic methods, like the Diffie-Hellman key exchange, as shown earlier. Therefore, with this paper, we encourage further development and usage of ECC in cloud computing security.

References

[1] National Institute of Standard and Technology: *The NIST definition of cloud computing*, `Web:http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf`. 2011.

[2] Tirthani, Ganesan: *Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography.* `Web:https://eprint.iacr.org/2014/049.pdf`. 2014.

[3] M. Knight: *The Merits of Elliptic Curve Cryptography.* `Web:https://www.thales-esecurity.com/blogs/2012/november/the-merits-of-elliptic-curve-cryptography`. 2012.

[4] A.H. Koblitz, N. Koblitz and A. Menezes: *Elliptic Curve Cryptography: The serpentine course of a paradigm shift.* `Web:http://math.boisestate.edu/~liljanab/MATH308/Literature/ECCKoblitz.pdf`. 2008.

[5] C. K. Koc: *Cryptographic Engineering.* Springer 2009.

[6] D. Hankerson, A. Menezes, S. Vanstone: *Guide to Elliptic Curve Cryptography.* Springer. 2004.

[7] Z. J. Shi, H. Yan: *Software Implementations of Elliptic Curve Cryptography.* `Web:http://www.engr.uconn.edu/~zshi/publications/shi08software.pdf`

[8] A. Menezes, M. Qu, D. Stinson, Y. Wang: *Evaluation of Security Level of Cryptography: ECDSA Signature Scheme.* `Web:https://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1051_ecdsa.pdf` 2001.

[9] M.H. Chakravarthy, E. Kannan: *Elliptic Curve Cryptography-Overview for Recent Cloud Architecture.* `Web:http://www.ijarcst.com/doc/vol2-issue4/ver.2/hemanth.pdf` .