

Proxy Oriented Identity-Based Data Uploading and Remote Data Integrity Checking in Public Cloud

^[1]M. Damodhar, ^[2]S.Poojitha

^[1]Dept. Of CSE, SVUCE, S.V.University, Tirupati., ^[2]Dept. Of CSE, SVUCE, S.V.University, Tirupati.

^[1]mdr.damodhar@gmail.com, ^[2]poojitha.singu@gmail.com

Abstract— Now a day's a number of clients were liked to store their data in public cloud along with the fast development of cloud computing. Novel security problems need to be resolved in order to facilitate processing of data for numerous clients over public cloud. Whenever a client is limited to access PCS (Public Cloud Services), he will hand over its proxy for his data process later upload them. Furthermore, remote checking of data integrity is an important security problem in public cloud. It provides the customers for checking that whether their data that is outsourced are kept undamaged without the requirement of downloading entire data. From the security problems, we propose an enhanced proxy-oriented data uploading and a model for remote data integrity checking over identity-based public-key cryptography: proxy-oriented identity-based data uploading and remote data integrity checking on public cloud. We provide a formal definition, system model as well as a model for security. Later, a concrete P-IDURIC protocol is designed with the help of bilinear pairings. The projected P-IDURIC protocol provides enhanced security based on hardness of the computational Diffie–Hellman problem. Furthermore the protocol is efficient and flexible. Depends on the actual client's authorization, the projected P-IDURIC protocol will realize delegated remote data integrity checking, private remote data integrity checking, and public remote data integrity checking.

Index Terms— Cloud computing, Public Cloud Services, Public key cryptography, remote data integrity checking

I. INTRODUCTION

Besides the rapid growth of computing and communication techniques, a vast deal of data is generated. These huge data requires more robust computation resources and extreme storage space. Over the past years, cloud computing fulfills the requirements of application and develops more quickly. Basically, it takes the data processing as a service, like storage, data security, computing, etc. With the utilization of public cloud platform, the clients were relieved from the problem of storage management, universal access of data along with independent geographical locations, etc. Hence, more number of clients would prefer to store and process their data with the help of remote cloud computing system.

In the environment of public cloud, the clients store their huge data over the remote public cloud servers. As the data stored is outside of the controlling capacity of clients, it involves the security risks for remote data in terms of integrity, confidentiality and availability services. Remote data integrity checking is the basic requirement that is useful to persuade the cloud clients that their data were preserved intact. In certain special cases, the owner of data may be limited in accessing the public cloud server, the data owner will delegate the job of data processing as well as uploading to the third-party, for example the proxy. On the opposite side, the remote data integrity checking protocol required to be efficient in order to make it best-suited for end devices with limited capacity. Hence, based on identity-based public cryptography and proxy public key cryptography, we will

learn P-IDURIC protocol.

A. Motivation

In the environment of public cloud, majority of clients uploads their own data to PCS and check their remote data's integrity via Internet. Whenever the clients are individual managers, certain practical problems will occur. If the manager is supposed of being involved due to the commercial fraud, he will be taken by the police. Throughout the investigation period, he will be restricted on accessing the network with the intention of guarding beside collusion. Nevertheless, the manager's legal business will move on throughout investigation period. Whenever a huge amount of data is produced, who can support him in processing this huge data? If such huge data cannot be processed with in time, the manager will entails loss of economic interest. In order to avoid this in happening, the manager is required to delegate the proxy to its data processing, for instance, his secretary. On the other hand, the manager will not expect that others have the capability to carry out the remote data integrity checking. Checking by public will entail several dangers over the leaking of privacy. For example, the data volume that is stored can be detected by the malicious verifiers. Whenever the data volume to be uploaded is confidential, private remote data integrity checking is essential. Even though the secretary has the capability to process and upload the data for the manager, he too cannot verify the manager's remote data integrity until he is permitted by the manager. We can say that the secretary as the proxy of a manager.

In PKI (public key infrastructure), remote data integrity

checking protocol will do the job of certificate management. Whenever the manager delegates certain entities to implement the remote data integrity checking, it involves significant overheads as the verifier can check the certificate meanwhile it verifies the remote data integrity. In PKI, the significant overheads emanate from the verification of heavy certificate, certificates generation, revocation, delivery, renewals, *etc.* In public cloud computing, the end devices can have less computation capacity like ipads, mobile phones, *etc.* Identity-based public key cryptography can eradicate the complicated certificate management. In order to improve the efficiency, proxy-oriented identity based data uploading and remote data integrity checking is quite attractive. Therefore, it will be most essential to learn about *P-IDURIC* protocol.

B. Related Work

There exist several diverse security problems in the cloud computing [1]. This paper depends upon the research results of proxy cryptography, identity-based public key cryptography and remote checking of data integrity in public cloud. In few cases, the cryptographic operation will be given to third party agents, for example proxy. Therefore, we need to make use of the proxy cryptography. Proxy cryptography is a extremely essential cryptography primitive. In 1996, Mambo *et al.* proposed the notion of the proxy cryptosystem [2]. During the instance that the bilinear pairings were brought into the identity-based cryptography, identity-based cryptography becomes efficient and practical. Since identity-based cryptography becomes more efficient because it avoids of the certificate management, more and more experts are apt to study identity-based proxy cryptography. In 2013, Yoon *et al.* proposed an ID-based proxy signature scheme with message recovery [3]. Chen *et al.* proposed a proxy signature scheme and a threshold proxy signature scheme from the Weil pairing [4]. By combining the proxy cryptography with encryption technique, some proxy re-encryption schemes are proposed. Liu *et al.* formalize and construct the attribute-based proxy signature [5]. Various other concrete proxy re-encryption schemes with their applications are also proposed [6]–[7].

E. Kirshanova proposed a unidirectional proxy re-encryption scheme which depends on the hardness of the LWE (Learning-with-Errors) problem. His construction is safe over collusion without the necessity of any trusted authority for the re-encryption key generation. In his paper he extended the trapdoor definition for a lattice of Micciancio and Peikert. The proxy re-encryption scheme is provides CCA-1 (Chosen Cipher-text Attack-1 usually known as lunch time attack) secure in the selective model under the assumption of LWE. But it does not provide CCA-2 security. Our proposed protocol is similar to this but we are not considering Lattices.

In the paper of Ohata *et al.*, the introduced a functionality for proxy re-encryption (PRE) known to be re-encryption verifiability. In their PRE scheme with re-encryption verifiability (which can be just call verifiable PRE, or VPRES), a receiver of a re-encrypted ciphertext is capable of verifying whether the ciphertext received is correctly transformed from

an original ciphertext through the help of a proxy, and therefore eventually detects illegal activities of the proxy. In their paper they formalized a security model for the scheme VPRES, and proved that the single-hop uni-directional PRE scheme by Hanaoka *et al.* (CT-RSA 2012) was extended to a secure VPRES scheme. Where there are limitations for security for public cloud services and it requires entire data need to be uploaded during checking and does not provide public key cryptography.

In public cloud, remote data integrity checking is the major security problem. As the massive data of clients is outside of their control, so the data may be corrupted by the malicious cloud server without the consideration of intentionally or unintentionally. In order to address this new security problem, several efficient and effective models are presented. In 2007, Ateniese *et al.* proposed a scheme known to be provable data possession (PDP) paradigm [8]. In his model, the checker checks the remote data integrity without the requirement of retrieval or downloading the entire data. PDP is a probabilistic proof of remote data integrity checking through sampling of random block sets from the public cloud server that radically decreases I/O costs. The checker performs the remote data integrity checking with the maintenance of tiny metadata. Later, various dynamic PDP model and protocols were designed [9]–[10]. Following Ateniese *et al.*'s pioneering work, various remote data integrity checking models and protocols have been proposed [11]–[12]. In 2008, proof of retrievability (POR) scheme was proposed by Shacham *et al.* [13]. POR is an efficient model that makes the checker not only check the remote data integrity but also it retrieves the remote data. Several POR schemes have been proposed [14]–[15]. In few situations, the client can delegate the remote data integrity checking work to the third party entities. In cloud computing, this auditing by third parties is indispensable [16]–[17]. Through the utilization of cloud storage clients can access the remote data from independent geographical locations. The end devices may be limited in computation and storage for example mobiles and handheld devices. Therefore, efficient and secure P-IDURIC protocol is best suitable for cloud clients equipped with mobile end devices.

From the role of the remote data integrity checker, all the remote data integrity checking protocols were classified into two different categories: private remote data integrity checking as well as public remote data integrity checking. In the response checking phase of private remote data integrity checking, some private information is indispensable. On the opposing, private information is not essential in the response checking of public remote data integrity checking. Especially, when the private information is given to the third party, then it can also perform the remote data integrity checking. In this situation, it is also known as delegated checking.

C. Contributions

In public cloud, this paper focuses on the proxy-oriented identity-based data uploading and remote data integrity checking. With the help of identity-based public key cryptology, our proposed P-IDURIC protocol is efficient because the certificate management is eliminated. P-IDURIC is a novel scheme of proxy-oriented data uploading and

remote data integrity checking scheme in public cloud. We provide the formal system model as well as security model for P-IDURIC protocol. Then, with the help of bilinear pairings, we designed a preliminary concrete P-IDURIC protocol. In the random oracle model, our designed P-IDURIC protocol is completely secure and can be proved. Based on the Actual client's authorization, our protocol can realize private checking, public checking and delegated checking.

D. Paper Organization

The paper is organized below. The formal system model and security model of P-IDURIC protocol are given in Section II. The concrete protocol, prototype implementation and performance analysis are presented in Section III. Section IV analyzes the proposed P-IDURIC protocol's security. The proposed P-IDURIC protocol is provably secure and efficient when compared with earlier models. At the end of the paper, the conclusion is given in Section V.

II. SYSTEM MODEL AND SECURITY MODEL OF ID-PUIC

In this section, we provide the system model and security model of P-IDURIC protocol. A P-IDURIC protocol consists of four different entities which are illustrated below:

- 1) **ActualClient**: an entity, which has large amount of data to be uploaded to PCS through the delegated proxy and also capable to perform the remote data integrity checking.
- 2) **PCS (Public Cloud Server)**: an entity that is managed by cloud service provider who has considerable storage space and computational resources for maintaining the clients' data.
- 3) **Proxy**: an entity that is authorized to process the ActualClient's data and upload them is selected and authorized by the ActualClient. When Proxy satisfies the warrant m_ω which is signed and issued by the Actual-Client, it is capable to process and upload the Actual client's data; otherwise, it cannot be able to perform the procedure.
- 4) **KGC (Key Generation Center)**: is an entity that on receiving an identity generates the private key that corresponds to the received identity.

In our proposed P-IDURIC protocol, ActualClient will interact with PCS for the checking of remote data integrity. Now, we give the formal definition and security model of P-IDURIC protocol.

Definition 1 (P-IDURIC): The P-IDURIC protocol is a combination of four phases (*Setup*, *Extract*, *Proxy-key Generation*, *TagGen*) and an interactive proof system (*Proof*). The details of these phases were described below.

- 1) **Setup**: When the security parameter k is input, the algorithm outputs the public parameters of the system and the master secret key. The system public parameters are made public and the master secret key msk is made confidential by KGC.
- 2) **Extract**: When the public parameters of the system, an identity ID , and the master secret key msk are given

as input, KGC outputs the private key sk_{ID} which corresponds to the identity ID .

- 3) **Proxy-Key Generation**: ActualClient generates the warrant m_ω and signs m_ω . Later it sends the warrant signature pair to the proxy. On receiving the warrant signature pair from ActualClient, the proxy generates the proxy-key with the help of its own private key.
- 4) **TagGen**: Input the file block F_i with the proxy-key, the proxy generates the corresponding tag T_i . Later it uploads the block-tag pair to PCS.
- 5) **Proof**: It is an interactive proof system between PCS and ActualClient. At the end of the interactive proof protocol, ActualClient outputs a bit $\{0 | 1\}$ to denote the "success" or "failure".

A practical P-IDURIC protocol must be efficient and provably secure. Based on the communication and computation overheads, efficiency analysis can be given. On the other side, a secure P-IDURIC protocol needs to satisfy the security requirements given below:

- 1) ActualClient can perform the P-IDURIC protocol without getting the local copy of the file(s) to be checked.
- 2) Only whenever the proxy is authorized, i.e., it satisfies the warrant m_ω , then only the proxy can process the files and uploads the block-tag pairs on behalf of ActualClient.
- 3) ActualClient cannot able to forge the proxy to generate block-tag pair's means that the proxy-protection property is satisfied.
- 4) If several challenged block-tag pairs are modified or lost, PCS's response cannot pass ActualClient's integrity checking.

III. THE PROPOSED P-IDURIC PROTOCOL

In this section, we propose an efficient P-IDURIC protocol to provide secure data uploading and storage service in public clouds. Bilinear pairings technique makes identity-based cryptography practical. Our P-IDURIC protocol is built on these bilinear pairings. We first review the bilinear pairings. Later the concrete P-IDURIC protocol is designed with the help of bilinear pairings.

A. Bilinear Pairings

Consider G_1 and G_2 as two cyclic multiplicative groups who have the same prime order q . Let Z_q^* denotes the multiplicative group of the field F_q . Bilinear pairings is a bilinear map $e : G_1 \times G_1 \rightarrow G_2$ [18] which satisfies the properties given below:

- 1) **Bilinearity**: $\forall g_1, g_2, g_3 \in G_1$ and $a, b \in Z_q^*$

$$e(g_1, g_2 g_3) = e(g_2 g_3, g_1) = e(g_2, g_1) e(g_3, g_1)$$

$$e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$$
- 2) **Non-degeneracy**: $\exists g_4, g_5 \in G_1$ such that $e(g_4, g_5) \neq 1_{G_2}$
- 3) **Computability**: $\forall g_6, g_7 \in G_1$, there is an efficient algorithm to compute $e(g_6, g_7)$.

The concrete bilinear pairings e can be constructed by using the modified Tate pairings [19] on elliptic curves. Our P-IDURIC protocol construction makes the use of the easiness of DDH (Decisional Diffie-Hellman) problem

whereas the security of our protocol is based on the hardness of CDH (Computational Diffie-Hellman) problem. Let g be a generator of G_1 . The CDH and DDH problems were defined below.

Definition 2 (CDH Problem on G_1): Given the triple $(g, g^a, g^b) \in G_1^3$ where $a, b \in Z_q$ are unknown, compute $g^{ab} \in G_1$.

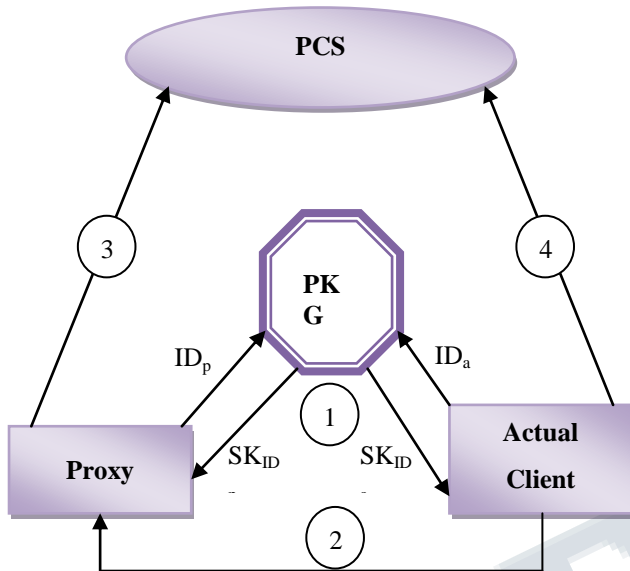


Fig.1. Architecture of our P-IDURIC protocol.

Definition 3 (DDH Problem on G_1): Given the quadruple $(g, g^a, g^b, g^c) \in G_1^4$ where $a, b \in Z^*_q$ are unknown, decide whether or not the formula $g^{ab} = g^c$ holds.

In the paper, we choose the group G_1 that satisfies the condition that CDH problem is quite difficult but DDH problem is simple. On the group G_1 , DDH problem is easy by making usage of the bilinear pairings. (G_1, G_2) are also known as GDH (Gap Diffie-Hellman) groups. On the groups G_1 and G_2 , the essential requirement is that the DLP (Discrete Logarithm Problem) is difficult. Let g_2 be a generator of G_2 . It is described below:

Definition 4 (DLP on G_1): Given (g, g_a) where $a \in Z^*_q$ is unknown, it is difficult to calculate a .

Definition 5 (DLP on G_2): Given (g_2, g_2^a) where $a \in Z^*_q$ is unknown, it is difficult to calculate a .

In the paper, we choose the groups G_1 and G_2 which satisfy that DLP, CDH are difficult while DDH is easy.

B. Our Concrete P-IDURIC Protocol

This concrete P-IDURIC protocol consists of four procedures: *Setup*, *Extract*, *Proxy-key generation*, *TagGen*, and *Proof*.

In order to show the intuition of our construction, the concrete protocol's architecture is depicted in Figure 1. First, *Setup* is performed later parameters of the system are generated. Depending upon the generated system parameters,

the remaining procedures are performed as shown in Figure 1. The steps were given below:

- 1) In the phase *Extract*, by giving the entity's identity as input, KGC generates the entity's private key. Particularly, it can generate the private keys for both client and the proxy.
- 2) In the phase *Proxy-key generation*, the actual client creates the warrant and helps the proxy in generating the proxy key.
- 3) In the phase *TagGen*, when the data block is input, the proxy generates the block's tag and upload block-tag pairs to PCS.
- 4) In the phase *Proof*, the actual client C' interacts with PCS. Through the interaction, C' checks its remote data integrity.

C. Performance Analysis

In terms of the computation and communication overhead of our proposed P-IDURIC protocol provides better performance when compared to other protocols. As our protocol mainly based on the group G_1 , bilinear pairings, exponentiation, and multiplication contribute most computation cost. Compared with them, the other operations are faster, such as, hash function h , the operations on Z^*_q and G_2 , etc. For the proxy, the computation overhead mostly comes from the *TagGen* phase. In the phase *Proof*, the actual client C' generates the challenge and PCS responds to this challenge. Since most computation cost comes from the pairings, exponentiation and multiplication on the group G_1 so our protocol takes less computation cost when compared to the Wang's protocol [10] for TagGen, PCS's cost in proof and checker's cost in proof. Also in that protocol there is no proxy data processing and uploading no integrity checking flexibility also it requires a certificate management.

National Bureau of Standards and ANSI X9 have been determined the shortest key length requirements for RSA and DSA is 1024 bits, ECC is 160 bits[20]. According to the above standard, we analyze that our P-IDURIC protocol's communication cost. After the data processing, the block-tag pairs are uploaded to PCS once and for all. Thus, we only consider the communication cost which is incurred in the remote data integrity checking. Hence our P-IDURIC protocol provides less communication cost when compared to others.

On the other hand, our protocol is capable of providing additional three security properties: proxy data processing and uploading, remote data integrity checking with flexibility and it does not require certificate management. Flexibility means that our protocol can realize the private checking, delegated checking and public checking according to the original client's authorization.

Hybrid Cloud: Our contributions are also suitable for the situation of hybrid clouds in which the proxy can be treated as the private cloud of the actual client. In such situation, the private cloud gets its private/public key pairs. The private cloud gets the proxy-key and the authorization of the actual client through the interaction between the actual client and its

private cloud. Whenever the actual client requires its private cloud to carry out the task of data uploading, it informs its private cloud. Upon receiving of the actual client's instruction, the private cloud interacts with the public cloud to finish the task of data uploading.

IV. SECURITY ANALYSIS

The security of our P-IDURIC protocol generally consists of the following parts: correctness, proxy-protection and unforgeability. The correctness has been shown in the subsection III-B. In the subsequent paragraph, we study the proxy-protection and unforgeability. Proxy-protection means that the actual client cannot pass himself off as the proxy to create the tags. Unforgeability means that when few challenged blocks are modified or deleted, PCS cannot send the valid response that is capable of passing the integrity checking.

V. CONCLUSION

Through the motivation of application requirements, this paper proposes the novel security model of P-IDURIC in public cloud. The paper formalizes P-IDURIC's system model and security model. Then, the first concrete P-IDURIC protocol is designed with the help of bilinear pairings technique. The concrete P-IDURIC protocol is provably secure and efficient by using the formal security proof and efficiency analysis. On the other hand, the proposed P-IDURIC protocol can also realize private remote data integrity checking, public remote data integrity checking and delegated remote data integrity checking based on the authorization of actual client's.

REFERENCES

- [1] Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Trans. Commun.*, vol. E98-B, no. 1, pp. 190–200, 2015.
- [2] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in *Proc. CCS*, 1996, pp. 48–57.
- [3] E.-J. Yoon, Y. Choi, and C. Kim, "New ID-based proxy signature scheme with message recovery," in *Grid and Pervasive Computing* (Lecture Notes in Computer Science), vol. 7861. Berlin, Germany: Springer-Verlag, 2013, pp. 945–951.
- [4] B.-C. Chen and H.-T. Yeh, "Secure proxy signature schemes from the weil pairing," *J. Supercomput.*, vol. 65, no. 2, pp. 496–506, 2013.
- [5] X. Liu, J. Ma, J. Xiong, T. Zhang, and Q. Li, "Personal health records integrity verification using attribute based proxy signature in cloud computing," in *Internet and Distributed Computing Systems* (Lecture Notes in Computer Science), vol. 8223. Berlin, Germany: Springer-Verlag, 2013, pp. 238–251.
- [6] E. Kirshanova, "Proxy re-encryption from lattices," in *Public-Key Cryptography* (Lecture Notes in Computer Science), vol. 8383. Berlin, Germany: Springer-Verlag, 2014, pp. 77–94.
- [7] S. Ohata, Y. Kawai, T. Matsuda, G. Hanaoka, and K. Matsuura, "Re-encryption verifiability: How to detect malicious activities of a proxy in proxy re-encryption," in *Proc. CT-RSA Conf.*, vol. 9048, 2015, pp. 410–428.
- [8] G. Ateniese *et al.*, "Provable data possession at untrusted stores," in *Proc. CCS*, 2007, pp. 598–609.
- [9] E. Esiner, A. Küpçü, and Ö. Özkasap, "Analysis and optimization on FlexDPDP: A practical solution for dynamic provable data possession," *Intelligent Cloud Computing* (Lecture Notes in Computer Science), vol. 8993. Berlin, Germany: Springer-Verlag, 2014, pp. 65–83.
- [10] H. Wang, "Proxy provable data possession in public clouds," *IEEE Trans. Services Comput.*, vol. 6, no. 4, pp. 551–559, Oct./Dec. 2013.
- [11] H. Wang, "Identity-based distributed provable data possession in multi- cloud storage," *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 328–340, Mar./Apr. 2015.
- [12] H. Wang, "Anonymous multi-receiver remote data retrieval for pay-TV in public clouds," *IET Inf. Secur.*, vol. 9, no. 2, pp. 108–118, Mar. 2015.
- [13] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. ASIACRYPT*, vol. 5350, 2008, pp. 90–107.
- [14] D. Cash, A. Küpçü, and D. Wichs, "Dynamic proofs of retrievability via oblivious RAM," in *Proc. EUROCRYPT*, vol. 7881, 2013, pp. 279–295.
- [15] T. Ma *et al.*, "Social network and tag sources based augmenting collaborative recommender system," *IEICE Trans. Inf. Syst.*, vol. E98-D, no. 4, pp. 902–910, 2015.
- [16] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
- [17] Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and C.-J. Hu, "Dynamic audit services for outsourced storages in clouds," *IEEE Trans. Services Comput.*, vol. 6, no. 2, pp. 227–238, Apr./Jun. 2013.
- [18] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proc. ASIACRYPT*, vol. 2248, 2001, pp. 514–532.
- [19] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E84-A, no. 5, pp. 1234–1243, 2001.
- [20] C. Research. *SEC 2: Recommended Elliptic Curve Domain Parameters*. [Online]. Available: <http://www.secg.org/SEC2-Ver-1.0.pdf>, accessed 2015.