小型サーバー上でのIDベース暗号鍵生成センタの実装

丹羽祐介 †　　金岡晃 ‡　　岡本栄司 ‡

† 筑波大学大学院システム情報工学研究科
305-8573 茨城県つくば市天王台 1-1-1
niwa@cipher.risk.tsukuba.ac.jp

‡ 筑波大学システム情報系
305-8573 茨城県つくば市天王台 1-1-1
{kanaoka, okamoto}@risk.tsukuba.ac.jp

あらまし ペアリングを利用したIDベース暗号は任意のデータを公開鍵とすることが可能であり, 公開鍵に対応した秘密鍵は鍵生成センタ (KGC：Key Generation Center) より生成される. 従来のKGCはPKIにおけるCAと同様に権威 (Authority) として考えられていた. 本稿ではさらなるIBEの実用化に向けて, KGCをAuthorityとして扱うのではなく, エンティティの一つとしてユーザ側でKGCを所有する状況を考える. そしてこのコンセプトの元, 現実的なアプリケーションを考慮した上で, KGCの機能を小型のLinuxマシンに搭載させ, パフォーマンスの評価を行う.

Development of a Key Generation Center for Identity-Based Encryption on Pocket Server

Yusuke Niwa†　　Akira Kanaoka‡　　Eiji Okamoto ‡

†Guraduate School of System and Information Engineering, University of Tsukuba
1-1-1, Tennodai, Tsukuba, Ibaraki, 305-8573, Japan
niwa@ciper.risk.tsukuba.ac.jp

‡Faculty of Engineering, Information and Systems, University of Tsukuba
1-1-1, Tennodai, Tsukuba, Ibaraki, 305-8573, Japan
{kanaoka, okamoto}@risk.tsukuba.ac.jp

Abstract

A KGC of IBE is usually regarded as an authority like the CA in PKI. We consider the case in which many KGCs exist in an IBE environment to put IBE into practical use. Each user can generate keys and the KGC is not considered as an authority in our proposed environment. We consider the use cases of KGC, and call the result “Pocket KGC”. As a proof of concept, we develop a KGC on a small, low-resource Linux server, and evaluate its performance.

1 Introduction

Identity-based encryption (IBE), first proposed by Shamir [1], is different from traditional public key encryption schemes. Users of an IBE system have the freedom to choose the public data the will use to encrypt a message. In traditional schemes, a key generation center (KGC) has been regarded as an authority like the certificate authority (CA) in public key infrastructure (PKI). However, regarding KGC as an authority puts limits on how it can be used in practice. In fact, KGC in the existing IBE system is most reliable organization. If we consider the wide range of services, KGCs should be deployed easily. Existing IBE

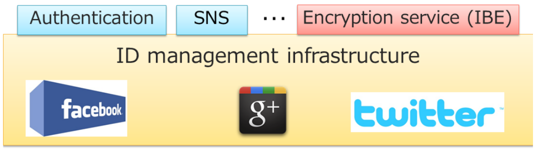

図 1: New Aspect of IBE

schemes handle a KGC as an authority. If we develop IBE system in existing way, operation of KGC might be so complex like interoperability between KGCs, trust domain, operation for revocation of keys and IDs, etc.

In this article, we introduce a new aspect of IBE. IBE is not infrastructure like conventional PKI, is just a one service on an existing identity management infrastructure (Figure 1). Moreover, KGCs are not authorities in this IBE service. Each KGC belongs to a given user and several KGCs coexist in the system. The new aspect of IBE gives new trust framework for IBE environment and it is easily implemented using existing identity management infrastructure, like organizational ID infrastructure or big ID issuing services (Ex. Facebook, Google, Yahoo!, Twitter).

IBE has a big advantage over conventional public key cryptosystems: the decryption keys can be generated after Bob receives the encrypted message from Alice. This advantage and the new aspect of IBE give various application fields, e.g electronic toll collection, smart meter, etc., which are difficult to realize with a traditional IBE system. In order to construct the applications described above, we take into account an implementation of a KGC with a lower-resource, which can be CPU, memory and size, works.

We consider a case in which many KGCs exist in an IBE environment. In this case, we call each KGC a "Pocket KGC", and consider that a Pocket KGC has two roles; *generation of user private keys* and *encryption of user messages* (Figure 2). In our IBE system, there are two entities, sender/encryptor with a Pocket KGC and receiver/decryptor, instead of three entities in traditional IBE systems. Then a sender/encryptor sends ciphertexts encrypted by a Pocket KGC to a receiver/decryptor. Authentication between a sender and a receiver is necessary in order to prevent man-in-the-middle attacks. The connection between the sender and the receiver must also be secure.

Similar systems can be realized using block ciphers. We have two advantages over this similar system. First, our proposed system does not need to store encryption keys, while this similar system has to store all encryption keys to respond for request from receiver/decryptor. We also need to consider how to share a common key between the sender and the receiver. These key managements are an essential point to construct a trust system. Furthermore, if there is little resource for storage in Pocket KGCs, our scheme is more effective. Second, our proposed system can assure that decryption keys are not generated before decryption key request. This advantage is one good property of IBE which generating decryption keys is not needed at the time of encryption.

We have to consider how we register, authenticate and manage IDs to develop and deploy existing IBE scheme. However, if we have our new aspect of IBE, we leave such ID management rolls to ID management infrastructure side. In other words, we can easily use other trust structures rather than construct own trust structure.

The rest of this paper is structured as follows. Section 2 describes state of the of the art solutions. In Section 3, we describe our motivation toward our proposal with concrete applications. In Section 4, we describe the implementation of Pocket KGC and evaluations.

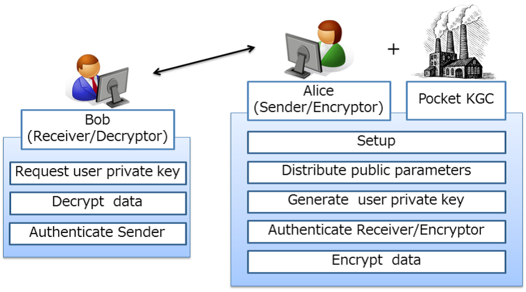

図 2: Proposal IBE System

2 Related Work

2.1 Identity-Based Encryption

An identity-based cryptosystem was first proposed by Shamir [1] in 1984. In 2000, Sakai, Ohgishi and Kasahara [3] proposed IBE with a pairing function. After them, Boneh and Boyen [4], Waters [5], and Gentry [6] have proposed advanced IBEs.

IBE has advantages for the management of certifications of public keys in that users can use identity-based data such as an e-mail address that identifies an individual person as public key information. In particular, no certifications are necessary in IBE, in contrast to traditional public key encryption, for which certifications are necessary to bind the public keys and their owners. However, a key generation center (KGC), which is a trusted third party, is needed to generate a user's secret key. For further details, we refer the reader to [7, 8].

In a IBE process, a KGC firstly discloses public parameters, and each sender encrypts a message with the public parameter and the sender's or receiver's ID. After receiving a ciphertext, a receiver can decrypt the ciphertext with his/her own ID by obtaining his/her secret key from the KGC.

2.2 Multipurpose IBE System

The multipurpose IBE system proposed in [9] is a communication specification that extends a data format proposed in RFC 5408 [2]. In the multipurpose IBE system, issuing a secret key involves two steps: verification of the issuing request by authenticating a user and the KGC actually generates the secret key. Figure 2 shows a process of timed-release encryption (TRE) [10] on the multipurpose IBE system. In TRE, the time is regarded as an ID and is used to encrypt a message. If a user wants to decrypt the message, he/she sends a request about the time used for encryption. After getting this request, the RA verifies that the requested time is correct. If the time is correct, KGC generates the user's key and sends it to the user.

2.3 Requirements of Multiple KGCs

As the PKI system demonstrates, it is actually quite difficult to manage an IBE system with only one KGC. Hence, we assume that multiple KGCs exist for any practical realization [11].

The multiple KGCs are also necessary mathematically. We can select strength of key depending on services in conventional PKI. If we want to realize such selection of strength in an IBE environment, we have to deploy several KGCs depending on types of key strength. We explain this using the Boneh-Franklin (BF) scheme [12], which is a representative scheme of the IBE system. The BF system has four algorithms: *Setup, Key Extract, Encrypt, and Decrypt*. The following equation is the *Key Extract* algorithm, which generates a user's secret key from the user's own ID. ID in the equation represents user's identity information, and D_{ID} is the user's secret key.

$$D_{ID} = H(ID)^s,$$

where $s \in \mathbb{Z}_p$ is a master secret key of the KGC, and H is a map-to-point hash function which outputs an element of a multiplicative cyclic group $\mathbb{Z}$ with prime order p from any bit string. As described above, the secret key can be obtained by calculating a modulo exponentiation of the group with the user's identity information and a master secret key. Therefore, different parameters are needed in order to generate a secret key which has a different size without changing a multiplicative cyclic group, a curve, and prime p. That's why we need several KGCs. In all IBE schemes, a KGC cannot generate a secret key of a different size from the same parameters. Hence, it is desirable that we consider a design with multiple KGCs to utilize the system based on the standardization.

2.4 Implementation of KGC

An IBE system requires a KGC as a trusted authority. A KGC has several roles: generation of public parameters, disclosure, generation of a master key, and generation of user's private key for his/her ID. A public parameter or a secret key is different for each IBE protocol or security level. Hence, we need each KGC to have all individual parameters. Several researchers have proposed practical solutions suitable to our requirements. For example, Kanaoka et al. [13] have implemented a KGC with Boneh-Boyen (BB1) scheme [4] following the data format given in RFC 5408 [2].

3 Motivation, Assumption

As we mentioned in Section 1, regarding KGC as an authority puts limits on how it can be used in IBE. In particular, a KGC is required to operate extremely rigorous in traditional IBE system. In this environment, it is difficult to construct flexible trust systems with IBE. In order to deploy the trust systems with IBE for various applications, we do put the trust point on identity infrastructures and consider IBE as one of the services of identity infrastructures. Then we consider the case in which many KGCs exist in an IBE environment. We do not regard KGCs as authorities hereafter. Several applications can take advantage of our proposal: electronic toll collection, smart meters, etc.

3.1 Electronic Toll Collection (Milo system)

Milo [14] consists of three main players: the driver, represented by an on-board unit (OBU); the company operating the OBU (abbreviated TSP, for "toll service provider"); and finally the local government (or TC, for "toll charger") responsible for setting the road prices and collecting the final tolls from the TSP, as well as for ensuring fairness on the part of the driver. The interactions between these parties are represented in Figure 3. Our Pocket KGC can play the role of the OBU. One can check that the system satisfies the following properties:

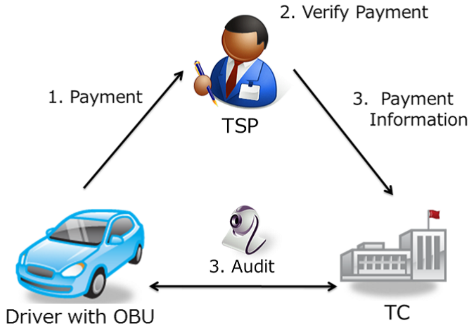

図 3: Concept of Milo

- **Driver privacy**
  Drivers should be able to keep their locations completely hidden from any other adversary, who may want to intercept (and

possibly modify) their payment information on its way to the TSP.

- **Driver honesty**
  Drivers should not be able to tamper with the OBU to produce incorrect location or price information. This property should hold even if drivers are colluding with other dishonest drivers, and should in fact hold even if every driver in the system is dishonest.

3.2 Smart Meter

A smart meter, which is one important function to realize a smart grid, is a system for the visualization of electric power distribution between consumers and electric power companies. The consumers can have a better sense in real time of their energy consumption and their electric-generating capacity, for example, in the form of solar power, and send this data to the electric power company via the electricity supply network itself. As a principle, private information must be kept secret: the amount of energy consumed is for instance a private information. Actually in order to satisfy confidentiality, some researchers consider the protocol that apply an Identity-based cryptography to smart meter in the AMI (Advanced Metering Infrastructure) [15]. The AMI is responsible for collecting, analyzing, storing, and providing the metering data sent by the smart meters.

Therefore, a new function based on generating a user's secret key is required. A smart meter can encrypt each entry recording the amount of energy consumed with some process name (or process ID) corresponding to an encryption key. Because the amount of energy consumed is encrypted with a process name, anyone wanting to decrypt this information needs to obtain a secret key from the smart meter. Naturally, a user can also browse their own energy consumption information as the owner of the smart meter (KGC). If each energy company charges for consumed energy, the user sends all his/her consumption data, which is encrypted, and sends the secret key for an appropriate process name (or process ID) generated by the smart meter (KGC). By doing this, the user can protect his/her privacy by avoiding disclosing inessential data.

4 Implementation of Pocket KGC

In this section, we describe our implementation of a pocket KGC.

4.1 Implementation on Low-Cost Linux server

We develop a KGC on a small, low-resource Linux server and call this KGC "Pocket KGC". Identity-based encryption is implemented according to the Boneh-Boyen (BB1) scheme [4]. Tables 1 and 2 summarize the main features of our development platform. The reason why we chose this server is because we can develop for versatile use with this server, which has rich external interface: two Network Interface and two USB ports. Although we may realize a KGC in smaller device depending on an application and required interfaces, we believe this server is reasonably small to try in various use cases.

表 1: Development environment

OS	SSD/Linux 0.5
CPU	AMCC PowerPC 405EX
CPU Frequency	600MHz
Memory	1GB
Software	Perl 5 PBC library 0.5.8 GMP library 5.0.2 OpenSSL 1.0.0a

表 2: Product Specification

Product Name	OpenBlockS 600
Figure Size	$81(W) \times 133(D) \times 31.8(H)mm$
Weight	265g

4.2 Evaluation of Performance for Issuing User Secret Key and Encrypting User Message

We evaluate the performance of the KGC, specifically, the calculation time required for the generation of a private key and the encryption of a message. We use the following parameters of the PBC Library [16]; *parameter a* from *type a*, *parameter d159, d201* and *d224* from *type d*, and *parameter e* from *type e*. Figure 4 and Figure 5 show each average processing time required for the private key generation and the encryption. From Figure 3, we see that the performance of *type a* (super singular curve) is the best choice to achieve a level of security of 80 bits (*param a, param d159* and *param e*).

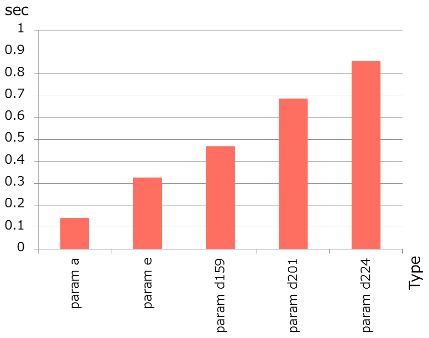

図 4: Average Processing Time for Private Key Generation

4.3 Development of Applications

We developed an application of key generation. This application supports RFC 5408 [2].

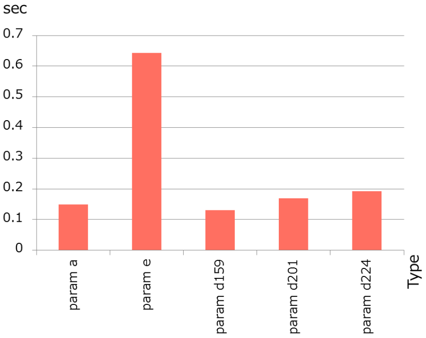

図 5: Average Processing Time for Message Encryption

To request a private key, a client must perform a HTTP POST method to Pocket KGC. The POST message contains XML data defined in RFC 5408. A <ibe:id> includes ibeIdentityInfo which encoded by Base64 of DER-encoded ASN.1 structure. If the key request is successful, the Pocket KGC responds with a responseCode of IBE100. Key generation primitive function is same as last subsection.

4.4 Implementation Results

In this subsection, we evaluate our KGC performance. The evaluation is focused on the processing time for an application to generate a private key as an XML file by using Apache benchmark a hundred times from the same subnetwork host. The processing time shown in Figure 6 is measured using several parameters of PBC: *param a* in *type a*, *param d159, param d201* and *param d224* in *type d*, and *param e* in *type e*. In this experiment, we do not consider high parallel access to Pocket KGC, because we consider communications between users and Pocket KGC is basically one-to-one. Namely, Pocket KGC in our proposal is different from the traditional IBE schemes, in the sense that Pocket KGC is not an authority anymore. Parallel accesses to the Pocket KGC are not frequent.

表 3: Average Processing Time

	param a	param e	param d159	param d201	param d224
Keygen (sec)	0.14	0.32	0.47	0.69	0.86
Encryption (sec)	0.15	0.64	0.13	0.17	0.19

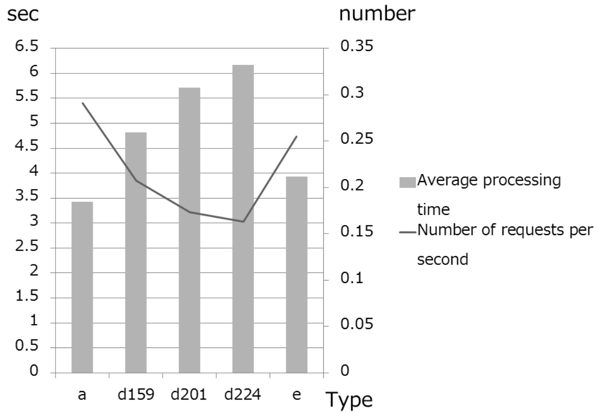

図 6: Average Processing Time for Application of Key Generation

5 Conclusion

In order to promote wide and flexible use of IBE, we do not regard KGCs as authorities, and consider that IBE is one of the services on existing identity infrastructures. It is possible to apply existing services, e.g. electronic toll collection and smart metering. As a proof of concept, we implemented a Pocket KGC providing the user with key generation and encryption on a low-resource Linux server. Our results show that our solution is acceptable for real-world applications.

参考文献

[1] A. Shamir. "Identity-based cryptosystems and signature schemes ", Volume 196 of LNCS, pp.47-53, CRYPTO 1984.

[2] G. Appenzeller, L. Martin and M. Schertler. "Identity-Based Encryption Architecture and Supporting Data Structures ", RFC 5408, IETF2009. `http://tools.ietf.org/html/rfc5408`.

[3] R. Sakai, K. Ohgishi and M. Kasahara. "Cryptosystems Based on Pairing", SCIS2000.

[4] D. Boneh and X. Boyen. "Efficient selective-ID secure identity based encryption without random oracles", Volume 3027 of LNCS, pp.223-238, EUROCRYPT 2004.

[5] B. Waters. "Efficient Identity-Based Encryption Without Random Oracles", Volume 3494 of LNCS, pp.114-127, EUROCRYPT 2005.

[6] C. Gentry. "Practical Identity-Based Encryption Without Random Oracles", Volume 4004 of LNCS, pp.445-464, EUROCRYPT 2006.

[7] G. Price, C.J. Mitchell. "Interoperation Between a Conventional PKI and an ID-Based Infrastructure", Volume 3545 of LNCS, pp.73-85, EuroPKI 2005.

[8] K.G. Paterson, G. Price. "A comparison between traditional public key infrastructures and identity-based cryptography", Volume 8, Number 3, pp.57-72, Elsevier, Inf. Secur. Tech. Rep. 2003.

[9] S. Oda, A. Nagai, G. Yamamoto, T. Kobayashi, H. Fuji, Y. Nakai, T. Matsuda and K. Matsuura."Design, Construction

and Implementation of a Multipurpose IBE", SCIS2010.

[10] J.H. Choen, N. Hopper, Y. Kim and I. Osipkov. "Provably Secure Timed-Release Public Key Encryption", ACM Trans. Inf. Syst, Secur., Volume 11, No.2, Article 8, pp.1-44.

[11] A. Kanaoka, M. Shimaoka and E. Okamoto. "Trust Framework for Identity-based Cryptography", IPSJ, Volume 51, No.9, pp.1692-1701, 2010.

[12] D. Boneh, and M. Franklin. "Identity-Based Encryption from the Weil Pairing", Volume 2139 of LNCS, pp.213-229, CRYPTO 2001.

[13] A. Kanaoka, T. Houri and E. Okamoto. "Achieving Identity-based Encryption Enabled SSL/TLS and Its Implementation on the OpenSSL and Apache HTTP Server", pp.1-13, JWIS 2011. https://sites.google.com/site/jwis2011/6A.pdf?attredirects=0

[14] S. Meiklejohn, K. Mowery, S. Checkoway and H. Shacham. "The Phantom Tollbooth: Privacy-Preserving Electronic Toll Collection in the Presence of Driver Collusion", pp.1-20, USENIX Security 2011.

[15] C. Bekara, T. Luckenbach and K. Bekara. "A Privacy Preserving and Secure Authentication Protocol for the Advanced Metering Infrastructure with Non-Repudiation Service", pp.60-68, ThinkMind ENERGY 2012.

[16] PBC Library http://crypto.stanford.edu/pbc/.