A Note on IBE Performance of a Practical Application

Ştefan Maftei
Faculty of Automatic Control
and Computers
University Politehnica of Bucharest
Bucharest, Romania
stefan_radu.maftei@stud.acs.upb.ro

Marius Supuran
Faculty of Automatic Control
and Computers
University Politehnica of Bucharest
Bucharest, Romania
marius.supuran@yahoo.com

Emil Simion
Faculty of Applied Sciences
University Politehnica of Bucharest
Bucharest, Romania
emil.simion@upb.ro

***Abstract*—Every user can be identified online by a unique string used for email or nickname on some of the many platforms out there. IBE systems propose a simple cryptosystem in which the public key system can be omitted by using the unique string as public identification. In this paper we present a minimal email application that uses Clifford Cocks' proposed IBE scheme. We analyze the impact of using it inside our application and how it can be improved to better fit the need of nowadays applications.**

***Index Terms*—IBE, cryptosystem, PKG, email, public key, private key**

I. Introduction

Identity-based encryption (IBE) refers to a type of cryptosystem where any string can be a public key - usually the string uniquely identifies the user (email address, social security number). The basic idea was first proposed by Adi Shamir in 1984 [1], however, concrete solutions have only arisen in recent years. The motivation behind the proposal was an attempt to simplify the infrastructure required for public key cryptography, avoiding the need for complex systems that store and disseminate public keys. A fully functional IBE system can easily include features such as automatic key expiration and revocation, inherent key escrow and "time capsule" messages (can only be read in the future).

After this brief introduction, we explain in section II what an IBE system consists of. Then in section III, we describe the types IBE schemes. Section IV presents an email application in which we integrated one of the previous described IBE scheme. The performance evaluation is done in Section V, and then we conclude this paper in Section VI.

II. Identity-based system

This section explains the basic functionality of an identity-based cryptosystem. Such a system contains the users (e.g. Alice and Bob) and a trusted third party: the Private Key Generator (PKG). Each member of the system has their own public and private keys:

- The **PKG**: generates its own pair of public and private keys. The public key is published, while the private key (also known as master key) is kept secret and used to generate other private keys.
- The **users**: a users' public key is either their identity ID (such as their email address) or can be derived from their identity ID using a *public process*. On the other hand, their private key is derived by the PKG using the identity ID and the master key, and then communicated to each user.

First, there are a number of preliminary steps. The PKG generates its pair of keys. Then, Alice and Bob each authenticate themselves to the PKG and obtain their private keys, based on their unique identity ID. Note that the communication between the user and the PKG is assumed to be over a secure channel, otherwise the private keys might be leaked.

When Alice wants to send a message to Bob, the following steps are taken:

1) Alice encrypts the message to Bob using either Bob's ID directly or a key generated from Bob's ID, which she can generate herself.
2) Alice signs the message using her own private key, obtained from the PKG.

Then, when Bob receives the message:

1) He uses Alice's public key (either the ID or derived from the ID) to verify the message authenticity.
2) He uses his own private key (obtained from the PKG) to decrypt the message, then he can read it.

III. State of the art

A. Elliptic Curves

The first fully functional IBE scheme to be developed was proposed by Boneh and Franklin [2] and based on elliptic curves. The scheme proposes four algorithms:

1) **Setup:** From some security parameters generates the *master-key* (private) and the system parameters (*params*), which are publicly known.
2) **Extract:** Has as input *params*, *master-key* and the ID - an arbitrary string used as public key. It outputs the private key d corresponding to the ID.
3) **Encrypt:** Taking as input *params*, ID and a message M, returns the encrypted ciphertext C.
4) **Decrypt:** Taking as input *params* and the private key d, it returns the plaintext message M.

The first two algorithms, **Setup** and **Extract**, can only be run on the PKG, as they require access to the *master-key*. **Encrypt** and **Decrypt** are run by the users in the IBE system.

The implementation of the scheme is based on bilinear maps between groups, such as the Weil pairing on elliptic curves. This assumes that a variant of the Computational Diffie-Hellman problem is hard.

This IBE scheme has *adaptive chosen ciphertext security*, assuming a *random oracle* model. Even more, the authors define and adversary more powerful than the standard adaptive chosen ciphertext model: the attacker can choose the ID (public key) to attack, and can obtain from the PKG the private key corresponding to any public key except the private key for the attacked ID. Even with these advantages the attacker still cannot obtain the desired private key.

B. Quadratic Residues

At about the same time as Boneh and Franklin, Clifford Cocks proposed another IBE scheme [3] based on *quadratic residues*. In this system, the PKG (called here the *authority*) generates two primes P and Q which represent the *master key* - P and Q must be both congruent to 3 mod 4. The product of P and Q is the modulus M and is publicly available, being equivalent to the system *params* from the Boneh-Franklin IBE scheme. The system also publishes a *public process* through which any string representing an ID can be transformed in a usable public key, this usually involves a secure hash function.

The system setup step is comprised of the generation of P and Q and the computation of M. Then, each time a user registers to the system, they present their identity to the IBE and are given the public modulus and their private key r, generated by the PKG.

When a user Bob (assume he knows M) wants to transmit an encrypted message to Alice, he uses M and Alice's ID to compute a, using the public process. Then he uses a and M to encrypt the message. Alice can then decrypt the message using her private key r and the modulus M.

The generation of a from a string ID is done by applying a hash function to the string repeatedly until the Jacobi symbol $(\frac{a}{M}) = +1$, and it can be done by any user. On the other hand, the generation of the private key r is based on the prime factors P and Q, which are known only to the PKG. The resulting private key satisfies either $r^2 \equiv a \bmod M$ or $r^2 \equiv -a \bmod M$.

In order to encrypt a message, each bit x is coded as either +1 or -1, then a value t is chosen such as the Jacobi symbol $(\frac{t}{M}) = x$. The value s is computed for each x as $s = (t + a/t) \bmod M$ - this is the encrypted x. The decryption can be achieved by computing $(\frac{s+2r}{M})$, recovering x.

There are a number of advantages and disadvantages of this scheme. The encryption and decryption steps are not computationally expensive, and the scheme is relatively easy to understand and implement. However, it suffers from a bandwidth issue: the ciphertext is very large. This comes from two sources:

- First, the sender cannot know (at least initially) if the receiver has the private key r which is the square root of a or $-a$, so it has to double down on the transmission, computing and sending the value for both options.
- The larger issues however is that each bit is encrypted as a number of size as large as M. This leads to massive ciphertext expansion which may or may not be acceptable depending on the situation.

C. Fuzzy IBE and attribute-based encryption

In 2005, Sahai and Waters [4] proposed two new concepts related to identity based encryption: fuzzy encryption and attribute based encryption.

Fuzzy encryption refers to an IBE system where the data encrypted with a public key based on identity *a* could be decrypted with a private key based on identity *a'*, as long as *a* and *a'* are close enough as measured by a certain metric. This comes in handy in a system where identity is based on biometrics, such as a fingerprint. The trouble with biometrics is that measurements can be noisy, thus two measurements of the same fingerprint may not be exactly identical, but close. The authors propose an IBE system where the identity can be derived from biometric data and be resilient to noisy data. From a theoretical point of view, basing the public identity key on biometric data in such a system makes perfect sense: this data is part of the identity of the user. The user can physically demonstrate ownership of this data to an authority and there's little to no risk of losing the public key. A leak of the biometric data is also not a major risk, as the data is used as the public key anyway.

Attribute based encryption refers to a system where a user can encrypt a document such that all users with some attributes could decrypt it. Suppose the identity of a system user in an organization is composed of a two attributes: department and name. If somebody wanted to send an encrypted message to everybody in a department, they would use the department name as the encryption key, and all users with that department attribute as part of their identity could decrypt the message. Such a system would also make use of fuzzy encryption, allowing multiple *close* keys (generated based on the attributes) to be used for encryption/decryption.

The implementation of fuzzy encryption is based on bilinear maps between groups, being somewhat similar to the Boneh-Franklin scheme. A user's private key is made of a set of private keys, each generated for one of the attributes. The system is secure under an adapted version of the Selective-ID security model, without using random oracles.

D. Other approaches and contributions

There are a number of research papers in the field of identity based encryption that deal with various issues of IBE or propose alternatives. One such issue is key revocation: in a classic IBE system, in order to revoke a user's key, one has to somehow include date information in the identity string, which can be hard to manage. One proposed solution [5] for fuzzy IBE systems reduces the revocation effort from linear to logarithmic in the number of system users, based on trees of attributes. The random oracle model used by the Boneh-Franklin model is often cited as a weakness of IBE systems, however, Boneh and Boyen [6] propose a hierarchical IBE system which is secure without random oracles under a slightly weaker security model called *Selective-ID*. Hierarchical IBE (HIBE) systems can also be constructed using lattices, as show in this paper [7]. This system is secure in the standard model of the learning with errors (LWE) problem.

IV. Email application overview

In the next parts we will focus on the Clifford Cocks' proposed IBE scheme [3] based on *quadratic residues*. We implemented a proof of concept email application which integrates the early mentioned IBE scheme, containing the Private Key Generator, the Mail Server and the client sending/checking options. The implementation was done using Python programming language considering its versatility and ease of use.

A. Private Key Generator

As presented before, a system using IBE must contain an entity responsible with private key generation for each user that participate to message exchanges. This entity is called **Private Key Generator** (referred next as **PKG**) which is an authority all users can trust.

The Private Key generation flow is straightforward:

1) A user connects to PKG.
2) The user demands a private key by sending the personal email as ID.
3) The PKG returns the private key to user and the public modulus M which will be used to get the hashed ID for message receivers.

In our email system, users requests their private keys either before sending an encrypted mail, or before checking their received emails in order to be able to decrypt the message. Our implementation of PKG uses the Clifford Cocks' proposed IBE-scheme.

B. Mail Server

The Mail Server component is a minimal custom SMTP running locally. It is responsible with the message delivery. It is part of the sending logic which creates an `inbox/` directory to store the processed emails.

C. Mail sending

Ahead of sending an email, users firstly request the private key (if they do not have it yet). This step is important because the encryption requires the receiver's hashed ID which is generated by knowing the public modulus M. Thus, the user can encrypt the mail's subject and body with the receiver hashed ID and simply forwarding it to the Mail Server. Locally, the encrypted message's subject and body are stored in the `inbox/` for the specified user, alongside a `JSON` file with the sender details and timestamp.

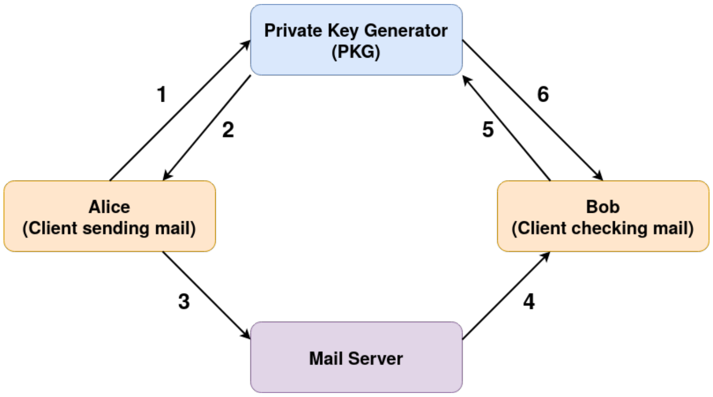

Figure 1. Example of sending and checking an email through the proposed email application.

D. Mail checking

Checking an email also requires the private key available which is obtained from the PKG authority if it was not generated prior to this. All received messages are encrypted by the sender using the hashed ID based on the receiver email. With the private key, a user can check the personal `inbox/` and decrypt the subject and body of each message. Then the whole email message is assembled and displayed to the user.

An example of how the email application works can be seen in Figure 1. Here Alice wants to send an email to Bob, both using the email application for the first time. Before sending, ALice needs to request a private key from the PKG in step 1, then in step 2 she receives it. Now Alice is ready to encrypt the email subject and body with Bob's hashed ID and send the email to him in step 3 through the Mail Server. Bob receives the email but before being able to check it in step 4, he must too get his private key from the PKG, sending the request in step 5 and reveiving it in step 6. Finally, Bob can check his inbox and read the email sent from Alice.

V. Email application performance

The performance of the email application focuses on how the IBE integrates within it and how big the encrypted messages are. As expected from [3] the system performs well computing the keys and encrypting the messages, but the main concern holds for the message sizes which can have a great impact upon the bandwidth.

We considered the case of a 2048-bit modulus M which for a 16 byte message would result in a 32 Kilobytes encrypted message, according to the original paper [3]. This is due to the encryption done for every single bit of the input data. To better visualize the encrypted data while being sent, we implemented the message as a string; therefore every digit of the resulted encryption is sent as a character (byte). The impact of this can be seen in Figure 2 against the theoretical data size sent raw.

With the big encrypted messages sizes in both cases described above, the Clifford Cocks' proposed IBE scheme is not suitable for all types of applications. If there is a constant exchange of messages, the user experience might be affected

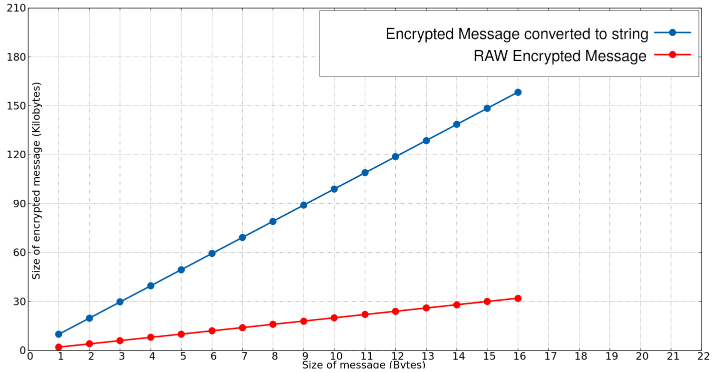

Figure 2. Encrypted data sizes with the IBE-scheme of an input message with sizes from 1 to 16 bytes with a 2048-bit Modulus.

if the bandwidth requirements are not met. However, to take advantage of the IBE cryptosystem it can be combined with other types of encryption such Advanced Encryption Standard (AES) [8], which is a symmetric block cipher standardized by NIST. Thus, the key can be sent using the IBE-scheme before a communication session or at an established time duration. The rest of the messages can be simply encrypted with AES, the messages sizes being comparably smaller than encrypted as before.

VI. Conclusions

In this paper we presented how Clifford Cocks' proposed IBE scheme works and how it can be integrated in an email application. Moreover, it can be used to in other applications as well if there is sensitive data or security parameters to be sent. The important advantage of the IBE scheme is the possibility of operating in an offline public key system.

Its disadvantage is the big encryption sizes that can have a tremendous impact upon the bandwidth, needing further investigations. One way of improving the performance can be achieved by combining the IBE scheme with AES encrypting only the key shared between the two users, taking advantage of the comparably small encryption sizes of AES compared to what we obtained using exclusively the IBE.

Another way of improving Cocks' proposed IBE scheme would be to encrypt at byte level instead of bit level to decrease the encryption size. This however is a difficult proposition which requires a function equivalent to the Jacobi which has byte-sized outputs. Alongside the problem of doubling the amount of encrypted data, these questions still remain unsolved.

References

[1] Shamir A. (1985) Identity-Based Cryptosystems and Signature Schemes. In: Blakley G.R., Chaum D. (eds) Advances in Cryptology. CRYPTO 1984. Lecture Notes in Computer Science, vol 196. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-39568-7_5

[2] Boneh D., Franklin M. (2001) Identity-Based Encryption from the Weil Pairing. In: Kilian J. (eds) Advances in Cryptology — CRYPTO 2001. CRYPTO 2001. Lecture Notes in Computer Science, vol 2139. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-44647-8_13

[3] Cocks C. (2001) An Identity Based Encryption Scheme Based on Quadratic Residues. In: Honary B. (eds) Cryptography and Coding. Cryptography and Coding 2001. Lecture Notes in Computer Science, vol 2260. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45325-3_32

[4] Sahai A., Waters B. (2005) Fuzzy Identity-Based Encryption. In: Cramer R. (eds) Advances in Cryptology – EUROCRYPT 2005. EUROCRYPT 2005. Lecture Notes in Computer Science, vol 3494. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11426639_27

[5] Boldyreva, Alexandra; Goyal, Vipul; Kumar, Virendra. (2008). Identity-based encryption with efficient revocation. 417-426. 10.1145/1455770.1455823.

[6] Boneh D., Boyen X. (2004) Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin C., Camenisch J.L. (eds) Advances in Cryptology - EUROCRYPT 2004. EUROCRYPT 2004. Lecture Notes in Computer Science, vol 3027. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-24676-3_14

[7] Agrawal S., Boneh D., Boyen X. (2010) Efficient Lattice (H)IBE in the Standard Model. In: Gilbert H. (eds) Advances in Cryptology – EUROCRYPT 2010. EUROCRYPT 2010. Lecture Notes in Computer Science, vol 6110. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13190-5_28

[8] FIPS PUB 197, Advanced Encryption Standard (AES), National Institute of Standards and Technology, U.S. Department of Commerce, November 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf