Analysis of Lewko-Sahai-Waters Revocation System

Zhengjun Cao[1], Lihua Liu[2,*]

Abstract

In 2010, Lewko, Sahai and Waters proposed an efficient revocation system but they neglected the security differences between one-to-one encryption and one-to-many encryption. In their system, an authority generates all users' decryption keys once and for all. We remark that the inherent drawback results in that the system is vulnerable to an attack launched by some malicious users. These malicious users could exchange their decryption keys after they receive them from the authority *in order to maximize their own interests.* Thus, the Lewko-Sahai-Waters revocation system cannot truly revoke a malicious user. From the practical point of view, the flaw discounts greatly the importance of the system.

Keywords. Broadcast encryption, revocation system, q-decisional multi-exponent bilinear Diffe-Hellman assumption.

1 Introduction

In 1991, Berkovits [1] introduced the primitive of broadcast encryption which was formalized by Fiat and Naor [7]. It requires that the broadcaster encrypts a message such that a particular set of users can decrypt the message sent over a broadcast channel. The Fiat-Naor broadcast encryption and the following works [8, 9, 13, 18, 19] use a combinatorial approach. This approach has to right the balance between the efficiency and the number of colluders that the system is resistant to. Recently, Boneh et al [2, 10] have constructed some broadcast encrypt systems. In these systems, the public parameters must be updated to allow more users.

In a revocation system, a broadcaster encrypts a message such that a particular set of revoked users cannot decrypt the message sent over a broadcast channel. In 1998, Kurosawa and Desmedt [14] introduced a method based on polynomial interpolation for constructing revocation systems. The subsequent revocation systems [17, 20] adopt this technique. In 1999, Canetti et al [3, 4] developed a different method for multicast encryption. In 2001, Naor, Naor and Lopspeich [16] proposed a stateless tree-based revocation scheme. Their method was subsequently improved

[1] Department of Mathematics, Shanghai University, Shanghai, China. [2] Department of Mathematics, Shanghai Maritime University, Shanghai, China. * liulh@shmtu.edu.cn

by Halevy and Shamir [12], by Goodrich, Sun, and Tamassia [11], and by Dodis and Fazio [6]. In 2007, Delerabl*ée*, Paillier and Pointcheval [5] proposed a revocation system which does not need to modify decryption keys when the public parameters are updated.

In 2010, Lewko, Sahai and Waters [15] proposed a simple revocation system with very small decryption keys. In the scheme, the size of public and decryption keys is only a constant number of group elements from an elliptic-curve group of prime order. The authority generates all users' decryption keys once and for all. In this paper, we remark that the inherent drawback results in that the system is vulnerable to an obvious attack launched by some malicious users. In order to maximize their own interests, these malicious users could conduct mutually beneficial cooperations by exchanging their decryption keys after they receive them from the authority. In addition, we will show that their security argument for the system is flawed.

2 Preliminaries

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order p, g be a generator of $\mathbb{G}$. A bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ has the following properties: (1) for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$; (2) $e(g, g) \neq 1$. We say that $\mathbb{G}$ is a bilinear group if the group action in $\mathbb{G}$ can be computed efficiently and there exists a group $\mathbb{G}_T$ and an efficiently computable bilinear map as above.

A revocation system is made up of three randomized algorithms [15]. An authority is responsible for running the setup algorithm which outputs a public key PK and master secret key MSK.

KeyGen(MSK, ID). The key generation algorithm takes in the master secret key MSK and an identity, ID. It generates a decryption key SKID for the identity.

Encrypt(S, PK, M). The encryption algorithm takes as input a revocation set S of identities along with the public key and a message M to encrypt. It outputs a ciphertext CT such that any user with a key for an identity ID $\notin S$ can decrypt.

Decrypt(S, CT, ID, D_{ID}) The decryption algorithm takes as input a ciphertext CT that was generated for the revocation set S, as well as an identity ID and a decryption key for it. If ID $\notin S$ the algorithm will be able to decrypt and recover the message M encrypted in the ciphertext.

q-Decisional Multi-Exponent Bilinear Diffe-Hellman Assumption. Let $\mathbb{G}$ be a bilinear group of prime order p. The q-MEBDH problem in $\mathbb{G}$ is stated as follows: A challenger picks a generator $g \in \mathbb{G}$ and random exponents $s, \alpha, a_1, \cdots, a_q$. The attacker is then given $\overrightarrow{y} =$

$$g, g^s, e(g,g)^\alpha; \quad \forall_{1 \le i,j \le q},\ g^{a_i}, g^{a_i s}, g^{a_i a_j}, g^{\alpha / a_i^2};$$

$$\forall_{1\leq i,j,k\leq q, i\neq j},\ g^{a_i a_j s}, g^{\alpha a_j/a_i^2}, g^{\alpha a_i a_j/a_k^2}, g^{\alpha a_i^2/a_j^2}.$$

It must remain hard to distinguish $e(g,g)^{\alpha s} \in \mathbb{G}_T$ from a random element in $\mathbb{G}_T$.

3 Lewko-Sahai-Waters revocation system

The construction uses a bilinear group $\mathbb{G}$ of prime order p. The identities are taken from the set $\mathbb{Z}_p$.

Setup. The authority picks random generators $g, h \in \mathbb{G}$ and exponents $\alpha, b \in \mathbb{Z}_p$ and publishes the public key $\mathrm{PK} = (g, g^b, g^{b^2}, h^b, e(g,g)^\alpha)$. The authority keeps α, b as secrets.

KeyGen(MSK, ID). For an identity ID, the authority picks a random $t \in \mathbb{Z}_p$ and returns the decryption key:

$$D_0 = g^\alpha g^{b^2 t}, \quad D_1 = (g^{b \cdot \mathrm{ID}} h)^t, \quad D_2 = g^{-t}.$$

Encrypt(PK, M, S). For a revocation set S of identities, let $r = |S|$ and ID_i denote the i-th identity in S. Pick random $s_1, \cdots, s_r$ and set $s = s_1 + \cdots + s_r$. Given a message M, it creates the ciphertext CT as:

$$C' = e(g,g)^{\alpha s} M, \quad C_0 = g^s$$

together with, for each $i = 1, 2, \cdots, r$:

$$\left(C_{i,1} = g^{b\, s_i}, \quad C_{i,2} = \left(g^{b^2\, \mathrm{ID}_i} h^b\right)^{s_i}\right).$$

Decrypt(S, CT, ID, D_{ID}). If ID $\in S$, then abort. Otherwise compute

$$\frac{e(C_0, D_0)}{e\left(D_1, \prod_{i=1}^{r} C_{i,1}^{1/(\mathrm{ID}-\mathrm{ID}_i)}\right) e\left(D_2, \prod_{i=1}^{r} C_{i,2}^{1/(\mathrm{ID}-\mathrm{ID}_i)}\right)}$$

which gives $e(g,g)^{\alpha s}$, this can be used to recover the message M from C'.

4 Security argument of Lewko-Sahai-Waters revocation system

In the section B.2 of Ref. [15], the authors presented a security argument for their revocation system. For convenience, we now relate it as follows.

Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage $\epsilon = \mathrm{Adv}_{\mathcal{A}}$ in the selective security game against our construction. Moreover, suppose attacks our system with a ciphertext of at most q revoked users. We show how to build a simulator, $\mathcal{B}$, that plays the decisional q-MEBDH problem.

The simulator begins by receiving a q-MEDDH challenge $\overrightarrow{X}, T$. The simulator then proceeds in the game as follows.

Initialization The adversary $\mathcal{A}$ declares a revocation set $S^* = \mathrm{ID}_1, \cdots, \mathrm{ID}_{r^*}$ of size $r^* \leq q$ that he gives to the simulator. (If $r < q$ the simulator will just ignore some of the terms given in $\overrightarrow{X}$).

Setup The simulator now creates the public key PK and gives $\mathcal{A}$ the private keys for all identities in S^*. The simulator first chooses a random $y \in \mathbb{Z}_p$ and sets $b = a_1 + a_2 + \cdots + a_r$. The public key PK is published as:

$$(g, g^b = \prod_{1 \leq i \leq r^*} g^{a_i}, g^{b^2} = \prod_{1 \leq i,j \leq r} (g^{a_i a_j}), h = \prod_{1 \leq i \leq r^*} (g^{a_i})^{-\mathrm{ID}_i} g^y, e(g,g)^{\alpha})$$

We observe that the public parameters are distributed identically to the real system and that the revocation set S^* is reflected in the simulation's construction of the parameter h.

Now the simulator must construct all private keys in the revocation set S. For each identity ID_i the simulator will choose a random $z_i \in Z_p$ and will set the randomness t_i of the i-th identity as $t_i = -\alpha/a_i^2 + z_i$.

The private key for ID_i is generated as follows:

$$\begin{aligned}
D_0 &= \left(\prod_{\substack{1 \leq j,k \leq n \\ \text{s.t. if}\, j=k \,\text{then}\, j,k \neq i}} (g^{-\alpha a_j a_k / a_i^2}) \right) \prod_{1 \leq j,k \leq n} (g^{a_j a_k})^{z_i} \\
D_1 &= \left(\prod_{\substack{1 \leq j \leq n \\ j \neq i}} (g^{-\alpha a_j / a_i^2})^{(\mathrm{ID}_i - \mathrm{ID}_j)} (g^{(\mathrm{ID}_i - \mathrm{ID}_j) a_j})^{z_i} \right) (g^{-\alpha / a_i^2})^y g^{y z_i} \\
D_2 &= g^{\alpha / a_i^2} g^{-z_i}
\end{aligned}$$

Challenge The simulator receives M_0, M_1 and chooses random $\beta \in \{0,1\}$. The simulator then chooses random $s', s'_1, \cdots, s'_r \in \mathbb{Z}_p$ such that $s' = \sum_i s'_i$. Let $u_i = g^{b^2 ID_i} h^b$. Note that this is computable from the public parameters, which were already set. The ciphertext will be encrypted under randomness $\tilde{s} = s + s'$ and be broken into shares $\tilde{s}_i = ais/b + s'_i$. Recall that $b = \sum_j a_j$. Therefore, $\sum \tilde{s}_i = \tilde{s}$. The challenge CT is created as

$$C' = Te(g,g)^{\alpha s'} M_\beta, C_0 = g^s g^{s'}, c_{i,1} = g^{s a_i} (\prod_j g^{a_j})^{s'_i}, c_{i,2} = \left(\prod_{\substack{1 \leq j \leq r^* \\ j \neq i}} (g^{s a_i a_j})^{\mathrm{ID}_i - \mathrm{ID}_j} \right) (g^{a_i s})^y u_i^{s'_i}.$$

Guess The adversary will eventually output a guess β' of β. The simulator then outputs 0 to guesses that $T = e(g,g)^{\alpha s}$ if $\beta = \beta'$; otherwise, outputs 1 to indicate that it believes T is a random group element in $\mathbb{G}_T$. When T is a tuple the simulator $\mathcal{B}$ gives a perfect simulation so we have that

$$\Pr[\mathcal{B}(\overrightarrow{X}, T = e(g,g)^{\alpha s}) = 0] = \frac{1}{2} + \mathrm{Adv}_{\mathcal{A}}.$$

When T is a random group element the message M_β is completely hidden from the adversary and we have $\Pr[\mathcal{B}(\overrightarrow{X}, T = R) = 0] = \frac{1}{2}$. Therefore, $\mathcal{B}$ can play the decisional q-MEBDH game with non-negligible advantage.

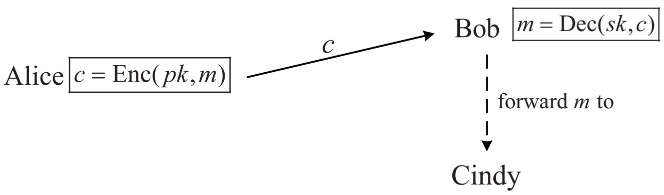
5 Different confidentiality levels

Confidentiality is a fundamental information security objective which is a service used to keep the content of information from all but those authorized to have it. From the sender's point of view, in a conventional one-to-one encryption the intended recipient undertakes the full obligations to keep the privacy of the plaintext. However, each intended recipient in a one-to-many encryption undertakes partial obligations. Based on this observation, we classify confidentiality into two kinds, strong confidentiality and weak confidentiality, corresponding to full obligations and partial obligations, separately. This classification will be helpful to analyze the behaviors of one intended recipient and revisit the security of different encryption models.

6 The security requirement for one-to-one encryption revisited

It is well known that the conventional one-to-one encryption requires that the adversary without the valid decryption key cannot recover the plaintext. Note that the adversary here is an uncharacteristic role. The requirement does not imply that some unintended recipients cannot recover or obtain the plaintext. In real life, some partners of the intended recipient can obtain or recover the plaintext by the following two methods.

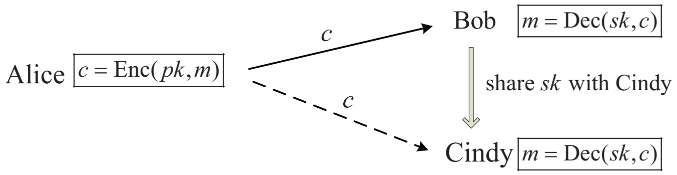
(1) The intended recipient, Bob, forwards the plaintext to his partner, Cindy. We refer to the Graph 1 for this case.



Graph 1: Bob forwards the plaintext m to Cindy

(2) The intended recipient, Bob, shares the decryption key with his partner, Cindy. We refer to the Graph 2 for this case.

In a word, the conventional one-to-one encryption has no intention to exclude some partners of the intended recipient from obtaining the plaintext. This property is so obvious that it is often neglected. However, the partnership of recipients must be taken into account when we



Graph 2: Bob shares his secret key with Cindy

design a one-to-many encryption system.

7 An inherent drawback in Lewko-Sahai-Waters system

Unlike a conventional one-to-one encryption, a broadcast encryption is a type of one-to-many encryption. Since there are many intended recipients, each recipient undertakes partial obligations to keep the privacy of the plaintext. Thus, an intended recipient possibly forwards the plaintext to others or shares his decryption key with others.

In a broadcast encryption system, a broadcaster encrypts a message such that a particular set of users can decrypt the message sent over a broadcast channel. It does not consider whether a valid user in the set reveals subsequently the message to others. We think, from the practical point of view, the security of a broadcast encryption consists in that an adversary can not obtain the message unless any valid user tells him/her.

In a revocation system, a user always expects to be able to decrypt any ciphertext even if he/she could be revoked in future. Thus, some malicious users could exchange their decryption keys in order to maximize their own interests. It is reasonable that they conduct mutually beneficial cooperations. For convenience, we call the attack decryption-key sharing. For example, Alice and Bob exchange their decryption keys after they receive them from the authority. Once Alice is revoked and Bob is not revoked, she shall use Bob's decryption key to decrypt any broadcasted ciphertext. Taking into account this attack, we remark that the Lewko-Sahai-Waters revocation system can not truly revoke a malicious user.

The inherent drawback is due to that the authority in the Lewko-Sahai-Waters system generates all users' decryption keys once and for all. The authors [15] neglected the partnership of recipients and paid less attentions to the security difference between a one-to-one encryption and a one-to-many encryption. By the way, the method to assign a fixed decryption key for each member is not applicable to revocation systems. Note that the Goodrich-Sun-Tamassia tree-based revocation system [11] is immune to the decryption-key sharing attack. They have stressed that keys should be updated after each insertion or deletion (revocation) of a device. They have also specified the strategy for key update and tree rebalance.

8 On the flawed security argument

8.1 On the inconsistent public keys

In the section B.2 of Ref. [15], the simulator sets the public key as

$$(g, g^b = \prod_{1\le i\le r^*} g^{a_i}, g^{b^2} = \prod_{1\le i,j\le r} (g^{a_i a_j}), h = \prod_{1\le i\le r^*} (g^{a_i})^{-\mathrm{ID}_i} g^y, e(g,g)^\alpha).$$

In view of that $g^b = \prod_{1\le i\le r^*} g^{a_i}$ and $g^{b^2} = \prod_{1\le i,j\le r}(g^{a_i a_j})$, we find $r^* = r$. By the way, the parameter r is not specified at all. This is a simple typo.

In the section 3.1 of Ref. [15], the authority sets the public key as

$$(g, g^b, g^{b^2}, h^b, e(g,g)^\alpha).$$

Note that the parameter h is not used by Encrypter, instead h^b. To keep the consistency of PK, it is better to set the public key in the simulation phase as

$$(g, g^b = \prod_{1\le i\le r^*} g^{a_i}, g^{b^2} = \prod_{1\le i,j\le r^*} (g^{a_i a_j}), h^b = \left(\prod_{1\le i\le r^*} (g^{a_i})^{-\mathrm{ID}_i} g^y\right)^{\sum_{j=1}^{r^*} a_j}, \ e(g,g)^\alpha).$$

We remark that the authors were not aware of the inconsistency. More worse, we find that the inconsistent PK is compatible with their subsequent security argument (see the phases of **Challenge** and **Guess**). That is, two different PK's are compatible with the same security argument. This seems against common sense.

In fact, the parameter h must be kept in secret. Otherwise, the adversary can launch the following equivalent-key attack. Concretely, a user with the identity ID can generate any equivalent keys which can be used for decryption. The user only need to pick a random $\phi \in \mathbb{Z}_p$ and compute

$$\begin{aligned}
\widehat{D_0} &= D_0 g^{b^2\phi} = g^\alpha g^{b^2(t+\phi)}, \\
\widehat{D_1} &= D_1 (g^{b\cdot \mathrm{ID}} h)^\phi = (g^{b\cdot \mathrm{ID}} h)^{t+\phi}, \\
\widehat{D_2} &= D_2 g^{-\phi} = g^{-(t+\phi)}.
\end{aligned}$$

Clearly, the new key $\{\widehat{D_0}, \widehat{D_1}, \widehat{D_2}\}$ can be used for decryption. Now the user reveals the equivalent key $\{\widehat{D_0}, \widehat{D_1}, \widehat{D_2}\}$ to the adversary without revealing the original decryption key $\{D_0, D_1, D_2\}$.

8.2 The simulator generates a false decryption key for ID_i

We now have a close look at the decryption key for ID_i which is generated by the simulator. Notice that

$$\begin{aligned} b &= a_1 + a_2 + \cdots + a_{r^*}, \\ t_i &= -\alpha/a_i^2 + z_i, \\ h &= \prod_{1\le i\le r^*} (g^{a_i})^{-\mathrm{ID}_i} g^y = g^{y-\sum_{1\le i\le r^*} a_i \mathrm{ID}_i}. \end{aligned}$$

Hence,

$$\begin{aligned} D_2 &= g^{\alpha/a_i^2} g^{-z_i} = g^{\alpha/a_i^2 - z_i} = g^{-t_i} \\ D_1 &= \left(\prod_{\substack{1\le j\le n \\ j\neq i}} (g^{-\alpha a_j/a_i^2})^{(\mathrm{ID}_i - \mathrm{ID}_j)} (g^{(\mathrm{ID}_i-\mathrm{ID}_j)a_j})^{z_i} \right) (g^{-\alpha/a_i^2})^y g^{y z_i} \\ &= \left(\prod_{\substack{1\le j\le n \\ j\neq i}} g^{a_j(-\alpha/a_i^2+z_i)(\mathrm{ID}_i-\mathrm{ID}_j)} \right) (g^{-\alpha/a_i^2+z_i})^y \\ &= \left(g^{\sum_{\substack{1\le j\le n \\ j\neq i}} a_j t_i (\mathrm{ID}_i-\mathrm{ID}_j)} \right) (g^{t_i})^y = \left(g^{\sum_{\substack{1\le j\le n \\ j\neq i}} a_j (\mathrm{ID}_i-\mathrm{ID}_j)} g^y \right)^{t_i} \\ &= \left(g^{y-\sum_{\substack{1\le j\le n \\ j\neq i}} a_j \mathrm{ID}_j + \mathrm{ID}_i \sum_{\substack{1\le j\le n \\ j\neq i}} a_j} \right)^{t_i} \end{aligned}$$

where n is specified as the quantity of queries to the group oracle made by the adversary (see the section B.1 in Ref. [15]). But the relation between n and r^* is not specified.

If $n = r^*$, then we have

$$\begin{aligned} D_1 &= \left(g^{y-\sum_{\substack{1\le j\le r^* \\ j\neq i}} a_j \mathrm{ID}_j + \mathrm{ID}_i \sum_{\substack{1\le j\le r^* \\ j\neq i}} a_j} \right)^{t_i} \\ &= \left(g^{y-\sum_{1\le j\le r^*} a_j \mathrm{ID}_j + \mathrm{ID}_i \sum_{1\le j\le r^*} a_j} \right)^{t_i} \\ &= \left(g^{y-\sum_{1\le j\le r^*} a_j \mathrm{ID}_j} g^{\mathrm{ID}_i \sum_{1\le j\le r^*} a_j} \right)^{t_i} = \left(h g^{b \mathrm{ID}_i} \right)^{t_i} \end{aligned}$$

$$
\begin{aligned}
D_0 &= \left(\prod_{\substack{1\le j,k\le r^* \\ \text{s.t. if}_{j=k}\,\text{then}_{j,k\neq i}}} (g^{-\alpha a_j a_k / a_i^2}) \right) \prod_{1\le j,k\le r^*} (g^{a_j a_k})^{z_i} \\
&= \left(\prod_{\substack{1\le j,k\le r^* \\ \text{s.t. if}_{j=k}\,\text{then}_{j,k\neq i}}} g^{a_j a_k} \right)^{-\alpha/a_i^2} \left(g^{\sum_{1\le j,k\le r^*} a_j a_k} \right)^{z_i} \\
&= \left(g^{\sum_{1\le j,k\le r^*} a_j a_k - a_i^2} \right)^{-\alpha/a_i^2} (g^{b^2})^{z_i} \\
&= \left(g^{b^2 - a_i^2} \right)^{-\alpha/a_i^2} (g^{b^2})^{z_i} \\
&= g^{\alpha} (g^{b^2})^{-\alpha/a_i^2 + z_i} = g^{\alpha} g^{b^2 t_i}
\end{aligned}
$$

The above D_0, D_1, D_2 constitute a proper decryption key for the identity ID_i.

If $n > r^*$ and $a_i = 0$ for $i = r^* + 1, \cdots, n$, then D_0, D_1, D_2 still constitute a proper decryption key for the identity ID_i.

If $n < r^*$, it is easy to check that D_0, D_1, D_2 constitute a false decryption key for the identity ID_i.

In sum, the simulation requires that the adversary have to make at lest (not at most) r^* queries to the group oracle. This restriction is of course against common sense. Usually, a simulation only specifies the upper bound to the quantity of queries made by the adversary.

One might argue that the number n in the representations of D_0 and D_1 is just a typo. It should be r^*. If that, we find the parameter n is not used in the whole simulation (see the four phases: Initialization, Setup, Challenge and Guess). Therefore, we do not know how n, the quantity of queries made by the adversary, exercises its influence on the advantage

$$\Pr[\mathcal{B}(\overrightarrow{X}, T = e(g,g)^{\alpha s}) = 0].$$

In other words, the representation of the advantage has no relation to the true quantity of queries. Therefore, it leads to a contradiction because the adversary can simply make less than r^* queries.

9 Conclusion

In this paper, we remark that the system is vulnerable to the decryption-key sharing attack. We also show that the security argument of Lewko-Sahai-Waters revocation system is flawed. From the practical point of view, we think it is unreasonable in a revocation system to ask the authority to generate all users' decryption keys once and for all.

References

[1] S. Berkovits, How to broadcast a secret, In Eurocrypt 1991, LNCS, vol. 547, pages 536-541. Springer-Verlag, 1991.

[2] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In CRYPTO 2005, LNCS, vol. 3494, pages 258-275. Springer-Verlag, 2005.

[3] R. Canetti, et al. Multicast security: A taxonomy and some efficient constructions. In Proc. of IEEE INFOCOM 1999, vol. 2, pages 708-716. IEEE, 1999.

[4] R. Canetti, T. Malkin, and K. Nissim. Efficient communication-storage tradeoffs for multicast encryption. In Eurocrypt 1999, LNCS, vol. 1592, pages 459-474. Springer-Verlag, 1999.

[5] C. Delerablée, P. Paillier, and D. Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In Pairing 2007, LNCS, vol. 4575, pages 39-59, 2007.

[6] Y. Dodis and N. Fazio. Public key broadcast encryption for stateless receivers. In Digital Rights Management Workshop 2002, pages 61-80. ACM, 2002.

[7] A. Fiat and M. Naor. Broadcast encryption. In Crypto 1993, LNCS, vol. 773, pages 480-491. Springer-Verlag, 1993.

[8] E. Gafni, J. Staddon, and Y.L. Yin. Efficient methods for integrating traceability and broadcast encryption. In Crypto 1999, LNCS, vol. 1666, pages 372-387. Springer-Verlag, 1999.

[9] J. Garay, J. Staddon, and A. Wool. Long-lived broadcast encryption. In Crypto 2000, LNCS, vol. 1880, pages 333-352. Springer-Verlag, 2000.

[10] C. Gentry and B. Waters. Adaptive security in broadcast encryption systems. In EUROCRYPT 2009, LNCS, vol. 5479, pages 171-188. Springer-Verlag, 2009.

[11] M. Goodrich, J.Z. Sun, and R. Tamassia. Efficient tree-based revocation in groups of low-state devices. In Crypto 2004, LNCS, vol. 3152, pages 511-527. Springer- Verlag, 2004.

[12] D. Halevy and A. Shamir. The LSD Broadcast Encryption Scheme. In CRYPTO 2002, LNCS, vol. 2442, pages 47-60. Springer-Verlag, 2002.

[13] R. Kumar, S. Rajagopalan, and A. Sahai. Coding constructions for blacklisting problems without computational assumptions. In CRYPTO 1999, LNCS, vol. 1666, pages 609-623, Springer-Verlag, 1999.

[14] K. Kurosawa and Y. Desmedt. Optimum traitor tracing and asymmetric schemes. In EUROCRYPT 1998, LNCS, vol. 1403, pages 145-157, Springer-Verlag, 1998.

[15] A. Lewko, A Sahai and B Waters: Revocation Systems with Very Small Private Keys. IEEE Symposium on Security and Privacy 2010, pages 273-285. IEEE, 2010. (Available at `https://eprint.iacr.org/2008/309`)

[16] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In Crypto 2001, LNCS, vol. 2139, pages 41-62. Springer-Verlag, 2001.

[17] M. Naor and B. Pinkas. Efficient trace and revoke schemes. In Proc. of Financial cryptog- raphy 2000, LNCS, vol. 1962, pages 1-20. Springer-Verlag, 2000.

[18] D. Stinson. On some methods for unconditionally secure key distribution and broadcast encryption. Des. Codes Cryptography, 12(3), 215-243. Springer-Verlag, 1997.

[19] D. Stinson and T. Trung. Some new results on key distribution patterns and broadcast encryption. Des. Codes Cryptography, 14(3), 261-279. Springer-Verlag, 1998.

[20] E. Yoo, et al. Efficient broadcast encryption using multiple interpolation methods. In Proc. of ICISC 2004, LNCS, vol. 3506, pages 87-103. Springer-Verlag, 2005.