Traitor Tracing Revisited: New Attackers, Stronger Security Model and New Construction

Xu An Wang[1] , Hailun Pan[2], Hao Liu[1], Xiaoyuan Yang[1]

[1]Engineering University of PAP
[2]Xi'an University of Posts and Telecommunications, P.R. China
wangxazjd@163.com

Abstract. In Crypto 94, Chor, Fiat, and Naor [8] first introduced the traitor tracing (TT) systems, which aim at helping content distributors identify pirates. Since its introduction, many traitor tracing schemes have been proposed. However, we observe until now almost all the traitor tracing systems using probabilistic public key (and secret key) encryption as the the content distribution algorithm, they do not consider this basic fact: the malicious encrypter can plant some trapdoor in the randomness of the ciphertexts and later he can use this trapdoor or the delegation of the trapdoor to construct decoding pirates, He can sell them to the black market and get his own benefits. At first sight, this new attack model is too strong to capture the real attack scenarios. But we think it is valuable at least for the following two reasons: (1) Note in many modern content distribution systems, there are at least existing three different roles: the content provider, the content distributer and the content consumer. In this framework, the encrypter is not necessarily the content provider (or content owner). He can be a malicious employee in the content provider corporation, it can also be the malicious content distributer corporation or its malicious employee. In all these cases, the encrypter has its own benefits and has the potential intention to plant some trapdoor in the randomness of ciphertexts. (2) Also note in the related work, there is a conclusion that traitor tracing and differential privacy can have directly influence on each other, while differential privacy (DP) is at the heart of constructing modern privacy preserving systems. But if we consider this new insider attacker (the encrypter), at least some part arguments on the relationship between traitor tracing and differential privacy in [9][18] need more consideration. Therefore in this paper we carefully describe this new insider attacker and investigate thoroughly on its effect. Our main research results are the following: (1) We show that many existing public key traitor tracing systems with probabilistic encryption algorithm are failing to work correctly when facing this malicious encrypter. They are including the BSW [4], BW [5], GKSW [11], LCZ [14] and BZ[6] traitor tracing systems. Furthermore, we conclude that most of the existing traitor tracing systems using probabilistic encryption algorithm can not resist this attack. (2) When considering the insider attacker (the encrypter), if the traitor tracing schemes using probabilistic encryption algorithms, the conclusion on tight relationship between traitor tracing and differential privacy shown in [9][18] may need more consideration. (3) By employing the technique of hash function, we show how to design TT^+ system which can resist this type of attack based on the existing traitor tracing system. Compared with the old traitor tracing system, our new proposal does not add much overhead and thus is practical too.

1 Introduction

Traitor tracing, introduced by Chor, Fiat, and Naor [8] in Crypto'94, mainly concerns on the following application: "Digital content provider (corporation) provides access control box which embedded some private keys to the paid consumers, later the paid consumers can decrypt the encrypted digital content by using the access control box and thus get their paid service. But a few of the consumers can be malicious and sell their access control boxes to the black market. Based on these boxes, some attackers with powerful technique can extract the private keys embedded in them, or they can construct new access control box which can also have the decryption ability. Traitor tracing is just aiming at stopping the happening of this situation. It adds a mechanism on the traditional broadcast encryption such that if the pirate boxes have been found or the running process of the pirate boxes can be observed, then they can be traced to the original malicious consumers".

Since traitor tracing have been introduced, many work have been done around this topic. Roughly speaking, traitor tracing can be categorized as many kinds. According to the broadcaster's key whether remain secret or public, there are public or secret BK TT systems. According to the trace key whether need be kept private or be public, there are public or secret tracing TT systems. According to the ability of collusion resistant, there are t-collusion resistant or full-collusion resistant TT system. According to treating the pirate decoder as a black-box oracle not, there are black-box tracing or white-box tracing TT system. According to the decoder is stateless or not, there are stateful or stateless TT system.

However an easily neglect fact about TT system is this: "if we treat the TT system as a protocol running by the digital content provider, the encrypter, the decrypter, the tracer and the pirate, then we do not consider a new type of attacker: the malicious encrypter. What shall happen if the encrypter is a malicious party? The current TT systems maybe can not work in this situation". Furthermore, the malicious encrypter model is reasonable in the practical broadcast encryption system for the encrypters are not always consistent with the digital content provider. For example, for BBC corporation, it is a digital content provider. But the holders of BBC corporation do not encrypt the digital content themselves, the employees implement the encryption. Thus the following may happen: "The encrypter may plant some trapdoor in the randomness when encrypting the digital contents. On one hand, the paid valid users can decrypt the encrypted content by using the traditional control box embedded private keys; but on the other hand, the encrypter sell this trapdoor to the black market, which can be used to decrypt the encrypted content but can not be traced by using traditional TT mechnisam.

It is important to point out the encrypters can not be only the corporation's employees, other parties can also play the role of encrypter. In the hierarchal digital content distribution system, the proxy sellers or the distribution corporation in the middle hierarchal also can encrypt the digital contents to the paid users, but they also can implement this trapdoor and sell it to the black market to gain economic benefits. It is also need to point out although the encrypter know the digital contents when encrypting, but if he directly leak these contents to others, he will be caught with high probability. But if he embed the trapdoor during the encryption and then sell this trapdoor to the black market, then the probability of being caught will be very low.

1.1 Our contribution

1. The traditional TT system will no longer work when facing this new attack, the existing TT systems have not considered this kind of attack when designing, thus they all fail to catch the real traitors (the encrypter). Some well known public key TT systems which can no longer work correctly are listed in the Table 1.
2. In 2009, Dwork et al.[9] for the first time show an interesting dual relationship between the TT system and the famous DP mechanism. This theoretical result give a very impressive tightly connection between TT and DP, and some computational complexity results on DP have been established on the optimal results in TT systems. But this optimal result maybe no longer hold in the stronger attacker model, thus we need reconsider these computational complexity results on DP carefully.

Table 1. Some well known public key TT systems can not work correctly in this stronger model

Schemes	Publication	Can not work correctly?
BSW06[4]	Eurocrypt 2006	Yes
BW06[5]	CCS 2006	Yes
GKSW10 [11]	CCS 2010	Yes
LCW13 [14]	CCS 2013	Yes

Table 2. Some computational complexity results on DP maybe need reconsider

The computational complexity results	Publication	Need reconsidered
DNRRV09[9]	STOC 2009	Yes
Ullman13[18]	STOC 2013	Yes
BZ14 [6]	Crypto 2014	Yes

3. We carefully search the ways to resist this new attack, by employing the technique of using hash function for generating randomness for encryption, we design TT^+ system which can resist this type of attack based on the existing traitor tracing system. Compared with the old traitor tracing system, our new proposal does not add much overhead and thus is practical too.

1.2 Related Work

The TT system can be categorized in different kinds according to their different features. Below we describe the related work following these categorization.

- According to the broadcaster whether using the public key encryption or symmetric encryption when broadcasting the digital content, the TT system can be regarded as public key kind or symmetric key kind. The first kind are very suitable for no setup applications, any one can complete the broadcast encryption, several well known results on this kind are [4][5] [11][14] [6], we will pay good attention on them, for almost all of them are probabilistic and it is possible to embed “trapdoor” in the randomness used in encryption. The second kind is suitable for practical digital media distribution system such as AACS system[1], most of the TT systems based on the [composite] technique are belonging to this kind, such as [16][3], but they also can be categorized as the first kind if the underlying symmetric encryption can be substituted as public key encryption, thus it is also possible to be attacked by malicious attacker.
- According to the different construction ways, the TT system can be regarded as the algebraic kind, the composite kind and the encoding/decoding kind. The first kind mainly based on the standard prime order group like DL-type group, the composite order group, the group with bilinear map and group with multilinear map etc, many of the public key TT schemes are belonging to this kind. The second kind mainly control the broadcasting, tracing and revocation by assigning different private keys to different users, the classic TT schemes constructed by using the “complete subset cover” technique or the “subset difference” technique are belonging to this kind. The third kind adapts techniques from encoding/decoding to implement the broadcasting, tracing and revocation mechanism, the well-known schemes such as [17] are belonging to this kind.
- According to whether the tracing key needed to be private or public, the TT system can be categorized as the private tracing kind and the public tracing kind. Most of the TT systems

belongs to the first kind. Compared with the first kind, the latter kind can complete the tracing without any secret, obviously this kind is more difficult for construction than the first kind such as [5] [7].

- According to the collusion resistant ability, the TT system can be categorized as the partial collusion resistant kind and fully collusion resistant kind. The first kind can only resist collusion attack by part of the users, while the latter can resist collusion attack by all the users. In 2006, Boneh et al. [4] put the first fully collusion resistant public key TT scheme by using bilinear map, after that, many subsequent TT schemes have been proposed, such as [5] [11]. In 2014, Boneh et al. [6] proposed a very efficient TT scheme based on indistinguishable obfuscation [10].
- According to whether the decryption (access control) box of the user need to maintain the state, the TT system can be categorized as the stateless kind and the stateful kind. For the first kind, the decryption box has no secret state information while the latter maintain secret state information. The latter kind is more difficult to be constructed than the first kind. [13] discussed how to transform a stateless TT scheme to be a stateful TT scheme.
- According to whether the pirate decryption box need to be treated as a black box or not when implementing the tracing mechanism, the TT system can be categorized as the black-box kind and the white-box kind. For the first kind, the tracing algorithm only need to treat the pirate decryption box as a black box, without concerning on its internal structures. For the second kind, the tracing algorithm need to use the internal structure of the pirate decryption box. It is easy to see the first kind is more difficult to construct than the second kind. Most of the existing TT systems belong to the first kind.
- According to the settings the TT system can be applied, the TT system can be categorized as the traditional kind, the identity based kind and the attribute based kind. When applying the TT mechanism in the identity based setting or the attribute based setting, the identity based TT systems [2] and the attribute based TT systems [14] [15] can be constructed.

1.3 Organization

The rest of this paper is organized as follows. Section 2 describe the preliminary knowledge. Section 3 describes security models of the scheme. In section 4, we review some classic TT schemes and show why they can not work again when facing the new attack. In section 5, we design TT^{+} scheme which is secure in this stronger model. Section 6 proves the security of our scheme. In Section 7, we give the performance analysis. Nextly, We discuss its potential application in cloud TV (CTV) in section 8. In the last section, we give our conclusion and list some future work.

2 Preliminary

2.1 Background Knowledge

Bilinear Pairing. Denote G and G_T as the two finite cyclic groups with same order $n = pq$, note here p and q are different primes. There exists a efficient computable function $e : G \times G \to G_T$ with the following properties:

1. Bilinear:$e\left(u^a, v^b\right) = e(u, v)^{ab}, \forall u, v \in G, \forall a, b \in Z$, we define the exponent is module n;

2. Non-degenerate: $e(g, g)$ can generate group G_T with order n, for some $g \in G$ which is also a generator of G.

We specify that the notation G_p is used to represent a subgroup of order p and G_q is used to represent a subgroup of order q.

Cryptographic Hash Function. A hash function H has the following properties:

1. One-wayness: for $H(x) = y$, given y, the probability of finding x is negligible;
2. Weak collision resistance: the probability of finding a value x_2 $(x_2 \neq x_1)$, given $H(x_1)$, which satisfies $H(x_1) = H(x_2)$, is negligible;
3. Strong collision resistance: the probability of finding $x_1 \neq x_2$ and $H(x_1) = H(x_2)$ is negligible.

2.2 Complexity Assumptions

For our proposal is based on [4] and we use the same assumptions as in [4], they are Decision 3-party Diffie-Hellman Assumption, Subgroup Decision Assumption, Bilinear Subgroup Decision Assumption, interested readers can refer to [4] for concrete contents.

3 Definition and Security Models

For our proposal is based on [4], and we use the same definition and security models as in [4], interested readers can refer to [4] for more concrete contents. However, we introduce a new security model for resisting the attack of trapdoor of randomness, which is the following:

Game − Resist the trapdoor of randomness. In this game, the adversary is a malicious encrypter. Suppose the adversary has a set of private keys $K_1, \cdots, K_t$.

− **Setup** The challenger runs the $Setup_{LBE}$ algorithm and gives the adversary PK.

− **Challenge** The adversary uses his private key set to construct a randomness s^* embedded some trapdoor. The adversary sends it to the challenger. The challenger randomly chooses a randomness R and computes $H^* = H(R, s^*)$. The challenger gives H^* to the adversary.

− **Guess** The adversary returns a guess R' of R to C.

We define the advantage of adversaryas $Adv_R = \frac{1}{|Y|}$ which $|Y|$ is the size of the hash function image space.

4 Review of Classic TT Scheme in the Stronger Security Model

How the BSW TT system work when facing the malicious encrypter: The complete scheme of B-SW06 is detailed given in [4]. Because malicious encryptor attack when executing the encryption algorithm, we focus on the encryption algorithm. When the encrypter encrypts, the encryption algorithm should choose random $t \in Z_n$, $w_1, \cdots, w_m, s_1 \cdots, s_m \in Z_n$.

However, the malicious encrypter do not choose $t \in Z_n$, $w_1, \cdots, w_m, s_1 \cdots, s_m \in Z_n$ all randomly, but "plant some trapdoors" in these randomness.

He first chooses a PRG generator (F) with secret key $K_1, \cdots, K_x, \cdots, K_m, K_t$, he then chooses random $w_1, \cdots, w_m \in Z_n$ and computes

$$\tilde{C_y} = g^{w_y}$$

and compute

$$t = F(K_t, \tilde{C_y}), s_x = F(K_x, \tilde{C_y})(x = 1, \cdots, m)$$

and compute four ciphertext components $(R_x, \tilde{R_x}, A_x, B_x)$, $C_y = H_y^t h^{w_y}$ as above. when he later want to decrypt the ciphertext, he computes

$$t = F(K_t, \tilde{C_y}), s_x = F(K_x, \tilde{C_y})(x = 1, \cdots, m)$$

and compute

$$M = B_x / G_x^{ts_x}$$

How does the encrypter constructs the pirates? He can sell his secret key $K_1, \cdots, K_x, \cdots, K_m, K_t$ or part of the secret key, or using the delegation paradigm of Pseudorandom Functions introduced in [12] to the black market to construct the pirates.

Since the tracing algorithm is running by the authority, and with high probability,

$$t \neq F(K_t, \tilde{C_y}), s_x \neq F(K_x, \tilde{C_y})(x = 1, \cdots, m)$$

thus if the tractor tracing algorithm tracing on encrypter's pirates, it will get an empty set, which will make the traitor either consider the pirate is of no useful to help decrypt the broadcast content, or the pirate has been trained cleverly to stop decrypt the test ciphertext sent by the authority. In one word, the encrypter can escape from tracing.

5 Our New TT^+ Scheme

1. *Setup*$(N = m^2, 1^k)$ Take the number of users N and a security parameter as input. It first gets two random prime numbers p and q, and a bilinear group G of composite order n where $n = pq$. Next it chooses random generators g_p,$h_p \in G_P$ and $g_q, h_q \in G_q$ and then computes $g = g_p g_q, h = h_p h_q \in G$. It selects a cryptographic hash function:$H : \{0, 1\}^* \to Z_n$.
   Notice that the cryptographic hash function H can be constructed according to the standard cryptographic hash functions like SHA-1. In particular, given a string x of arbitrary length as the input of SHE-1 and thus gets a fixed length output, and then the fixed length string is converted into an element in Z_n.
   It selects random exponent $r_1, \cdots, r_m, c_1, \cdots, c_m, \alpha_1, \cdots, \alpha_m \in Z_n$ and $\beta \in Z_q$. The system public parameter PK is created as follows:

$$g, h, E = g_q^{\beta}, E_1 = g_q^{\beta r_1}, \cdots, E_m = g_q^{\beta r_m}, F_1 = h_q^{\beta r_1}, \cdots, F_m = h_q^{\beta r_m}$$

$$G_1 = e(g_q, g_q)^{\beta \alpha_1}, \cdots, G_m = e(g_q, g_q)^{\beta \alpha_m}, H_1 = g^{r_1}, \cdots, H_m = g^{r_m}$$

   For user (x, y), the private key is computed as $K_{x,y} = g^{\alpha_x} g^{r_x c_y}$.The tracing private key K contains p, q and values that are used in generating PK.
2. *TrEncryp*$(K, M, (i, j))$ The algorithm is the secret key algorithm of the tracing authority. It encrypts a message $M \in G_T$ to the user subset whose row value is larger than i or both row value is equal to i and column value is larger than or equal to j. It inputs K, a message $M \in G_T$ and an index i, j. It first chooses random values $\delta \in Z_n, \eta_1, \cdots, \eta_m, \gamma_1, \cdots, \gamma_m, z_{p,1}, \cdots, z_{p,j} \in Z_n$,$(v_{1,1}, v_{1,2} v_{1,3}), \cdots, (v_{i-1,1}, v_{i-1,2} v_{i-1,3}) \in {Z_n}^{(3)}$.
   The row ciphertext components $(R_x, \widetilde{R}_x, A_x, B_x)$ are created as follows:

if $x > i$,

$$R_x = g_q^{r_x\eta_x}, \widetilde{R}_x = h_q^{r_x\eta_x}, A_x = g_q^{\eta_x\delta}, B_x = (M||R)e(g_q, g)^{\alpha_x\eta_x\delta}$$

if $x = i$,

$$R_x = g^{r_x\eta_x}, \widetilde{R}_x = h^{\eta_x r_x}, A_x = g^{\eta_x\delta}, B_x = (M||R)e(g, g)^{\alpha_x\eta_x\delta}$$

if $x < i$,

$$R_x = g^{v_{x,1}}, \widetilde{R}_x = h^{v_{x,1}}, A_x = g^{v_{x,2}}, B_x = e(g, g)^{v_{x,3}}$$

The column ciphertext components $C_y, \widetilde{C_y}$ are created as follows:
if $y \geq j$,

$$C_y = g^{c_y t} h^{\gamma_y}, \widetilde{C_y} = g^{\gamma_y}$$

if $y < j$,

$$C_y = g^{c_y t} g_p^{z_{p,y}} h^{\gamma_y}, \widetilde{C_y} = g^{\gamma_y}$$

3. *Encryp*(PK, M) It is executed by encrypting a message M using an encrypter such that all the users can receive the ciphertext. It first chooses random R and both the encrypter and users agree in advance that the size of R is 64 bits, then it computes

$$\delta = H(M \parallel R), \eta_x = H(M \parallel R||x)(x = 1, \cdots, m), \gamma_y = H(M \parallel R||y)(y = m + 1, \cdots, 2m)$$

The row ciphertext components $(R_x, \widetilde{R}_x, A_x, B_x)$ are created as follows:

$$R_x = E_x^{\eta_x}, \widetilde{R}_x = F_x^{\eta_x}, A_x = E^{\eta_x\delta}, B_x = (M||R)G_x^{\eta_x\delta}$$

The column components $C_y, \widetilde{C_y}$ are created as follows:

$$C_y = H_y^{\delta} h^{\gamma_y}, \widetilde{C}_y = g^{\gamma_y}(y = 1, \cdots, m)$$

4. *Decryp*$((x, y), K_{x,y}, C)$ The user (x, y) uses his private key $K_{x,y}$ to decrypt the ciphertext sent by the broadcast by computing:
The user (x, y) utilizes his private key $K_{x,y}$ to decrypt the ciphertext sent by the broadcast by computing:

$$M||R = B_x(e(K_{x,y}, A_x)e(\widetilde{R}_x, \widetilde{C}_y)/e(R_x, C_y))^{-1}$$

Finally, the user (x, y) can get M by removing the last 64 bits.

6 Security Proof

[4] contains three basic security, including index hiding, indistinguishability and message hiding. Since our proposal is constructed on the basis, interested readers can refer to [4] for more concrete contents. Note that, unlike EC:BonSahWat06, the random numbers used by the encryption algorithm are generated by a secure hash functions $H : \{0, 1\}^* \rightarrow Z_n$ when proving these three basic security. Moreover, our scheme also implements the security for resisting the attack for trapdoor of randomness, which is proved as follows:

6.1 Proof of Security for Resisting the Randomness of Trapdoors

Lemma 1. Uniformity-Suppose that $F^{X,Y}$ is the set of all hash functions from set X to set Y,then we call it a hash family.We say that the family is uniform if all hash values are equally likely $P(H(x) = y) = \frac{1}{|Y|}$ for any hash values y.

The theorem 1 is concerned with the adversaries advantage in winning the game when the encryptor uses a hash function which is mapped from set X to set Y which has a size of $|Y|$ to generate random numbers for encryption.Because the hash results are evenly distributed, the results cannot be embedded in the trap. If it can be embedded, it may mean that the hash function used is not safe.

Lemma 2. Unipolarity: given a value y, the probability of finding a value x, where $H\left(x\right) = y$ is negligible.

When the encryptor use the hash function to generate a random number for encryption, M is involved in the operation, such as $\delta = H(M||R)$. Since the hash function is unidirectional, when δ is leaked, the attacker cannot get M either.

Lemma 3. Weak collision resistance: given $H(a)$, the probability of finding a value b where $b \neq a$ and $H(b) = H(a)$ is negligible.

The encryptor generates random groups of numbers and row ciphertext blocks $(R_x, \widetilde{R}_x, A_x, B_x)$:

$$\delta = H(M \parallel R), \eta_x = H(M \parallel R \parallel x)(x = 1, \cdots, m), \gamma_y = H(M \parallel R \parallel y)(y = m + 1, \cdots, 2m)$$

$$R_x = E_x^{\eta_x}, \widetilde{R}_x = F_x^{\eta_x}, A_x = E^{\eta_x \delta}, B_x = (M \parallel R)G_x^{\eta_x \delta}$$

If a malicious encryptor finds a trap door $\bar{R}$ to replace $M \parallel R$, then

$$\bar{\delta} = H(\bar{R}), \bar{\eta}_x = H(\bar{R} \parallel x)(x = 1, \cdots, m), \bar{\gamma}_y = H(\bar{R} \parallel y)(y = m + 1, \cdots, 2m)$$

When $\bar{\eta}_x = \eta_x, \bar{\delta} = \delta$, the malicious encryptor can computer$M \parallel R = B_a / G_a^{\bar{\eta}_a \bar{\delta}}$. Because the hash function has weak collision resistance, the random numbers generated by each of them are not the same if $\bar{R} \neq M \parallel R$ .We can say that $\bar{\eta}_x \neq \eta_x, \bar{\delta} \neq \delta$ always holds.

Lemma 4. Strong collision resistance: the probability of finding two values such that $b \neq a$ and $H(b) = H(a)$ is negligible.

If the two inputs of the hash function are different, the outputs are also different, which ensures that the $2m+1$ random numbers used for encryption are different, and the same ciphertext will not appear in each row and each column.

In addition, the attacker cannot find $\bar{M} \parallel \bar{R}$, where $\bar{M} \neq M, \bar{R} \neq R$,which makes $\bar{\delta} = \delta, \bar{\eta}_a = \eta_a$. If it can be found, the attacker can get the message by computing $M \parallel R = B_a / G_a^{\bar{\eta}_a \bar{\delta}}$. Since our hash function has strong collision resistance, the message can not be leaked in this way.

Theorem 1. All adversaries have advantage $\frac{1}{|Y|}$ in playing the resist the randomness of trapdoors game.

7 Performance analysis

In this section, we evaluate the functionality and performance of TT^{+} scheme.

7.1 Property analysis

This part of the section provides a functional comparison of our scheme with BSW06[4]. From the data in Table 1, it can be seen that both our scheme and BSW06[4] belong to the category of broadcast encryption, and both have traceability. However, BSW06[4] is not resistant to malicious encryptor attacks. That is to say, the encryptor in BSW06 can embed trapdoors in random numbers, which is a great threat to the confidentiality of the data.

Our TT^+ scheme uses a hash function to generate random numbers for encryption. Since the output of the hash function is evenly distributed, malicious encryptors cannot embed trapdoors in random numbers. Note that in many modern content distribution systems, there are at least three different roles: the content provider, the content distributor, and the content consumer. The encryptor can be a malicious employee of a content provider company, or he can be a malicious content distributor company or its malicious employees. In all of these cases, the encryptor may have an underlying intention of embedding some trapdoors into the randomness of the ciphertext to gain profit, so we think it would be very valuable to consider this new insider attacker.

Table 3. Functionality comparison.

Scheme	Broadcast encryption	Traceability	Resist malicious encryptors
BSW06[4]	Yes	Yes	No
Ours	Yes	Yes	Yes

7.2 Property analysiss

To evaluate the performance of our scheme, we compared it with BSW06EC:BonSahWat06 and the details of which are shown in Table 2. H represents the hash operation, P represents the bilinear pairing operation, S represents the power operation in G, F represents the power operation in G_p, E represents the power operation in G_q, I represents the power operation in G_T, N indicates the number of users and $N = m^2$.

It is obvious from Table 2 that the computational cost of encryption in our scheme is higher than that of BSW06[4]. In order to prevent malicious encryptors from embedding trapdoors in random numbers, we use a hash function to generate random numbers, which causes the encryption cost of our scheme to exceed BSW06[4]. However, other than the overhead of the hash operation, the other overhead of the encryption phase is the same as that of BSW06[4].

In terms of decryption, the proposed scheme and BSW06[4] have the same overhead, only three bilinear pairing operations. Similarly, our scheme has the same tracing overhead as BSW06[4]. Because the tracing algorithm is executed by the tracing authority, there is no need to use the hash function.

8 Cloud TV: A Case Study

Cloud TV (CTV) is a software platform which practically describes the function of a Set-Top-Box (STB). CTV enables pay television operators and other video service providers to provide

Table 4. Efficiency comparison.

Scheme	Encrypt	Decrypt	Trace
BSW06[4]	(3S+3E+I)m	3P	3(m+i)S+3(m-i)E+mI+mp+(j-1)F
Ours	(N+1)H+(3S+3E+I)m	3P	3(m+i)S+3(m-i)E+mI+mp+(j-1)F

advanced user functionality to online videos like YouTube and iQIYI. CTV is the place where content is swallowed, encoded, encoded and transmitted. Operators are able to manage customer settings, gain knowledge of particular customer segments, and offer specific content to users.

Digital content providers provide paying consumers with private keys, and paying consumers can use the private keys to decrypt the encrypted digital content and get their paid content. But some consumers may be malicious and will sell their private keys to the black market. Based on these private keys, some technically powerful attackers can build new access control boxes with decryption capabilities (pirate decoder). If a pirate decoder is found, or observed in operation, it can be traced back to the original malicious consumer using the traitor tracing scheme.

However, if the encryptor is a malicious party, the malicious encryptor can embed trapdoors in the random number, and sell trapdoors to the black market for profit. In order to prevent malicious encryptors from embedding trapdoors in random numbers, the encryption algorithm uses a secure cryptographic hash function H to generate random numbers during encryption.Therefore, using our TT^+ scheme for cloud TV, cloud service providers can securely deliver content from content providers to all paying users.

The diagram shown in Figure 1 represents a system architecture of TT^+ scheme for CTV. The framework consists of three essential components, namely content providers, the cloud service provider, paying users, and the tracing authority. The content providers provide the content to the cloud server provider that encrypts the content and sends the broadcast ciphertext to paying users. Paying users can securely decrypt the encrypted digital content and get their paid content using their own private keys.

However, malicious users can sell their private keys to attackers who can use them to build pirate decoder with decryption capabilities. If a pirate decoder is found, the tracing authority can execute tracing algorithm to find the original malicious consumer. The cloud service provider can then take legal action against the traitors.

In addition, employees of the cloud service provider may also engage in malicious behavior.As shown in Figure 2, malicious encryptors could embed trapdoors in random numbers when encrypting paid content, and then sell trapdoors to the black market for profit. Attackers use trapdoors to construct pirate decoder that ignore any relevant digital rights restrictions to extract content they would otherwise pay for. Even worse, attackers can make their pirate decoder widely available so that anyone can extract content for themselves. When detecting the pirate decoder, tracing authority can run the tracing algorithm to interact with the pirate decoder. Unfortunately, because the pirate decoder is constructed by using the trapdoors, the output of the tracing algorithm is empty set. That is to say it can not find the illegal elements, which seriously infringes the legitimate rights and interests of content providers and the cloud service provider.

We can use a secure cryptographic hash function H to generate random numbers in the encryption algorithm, which prevents malicious encryptors from embedding trapdoors in random

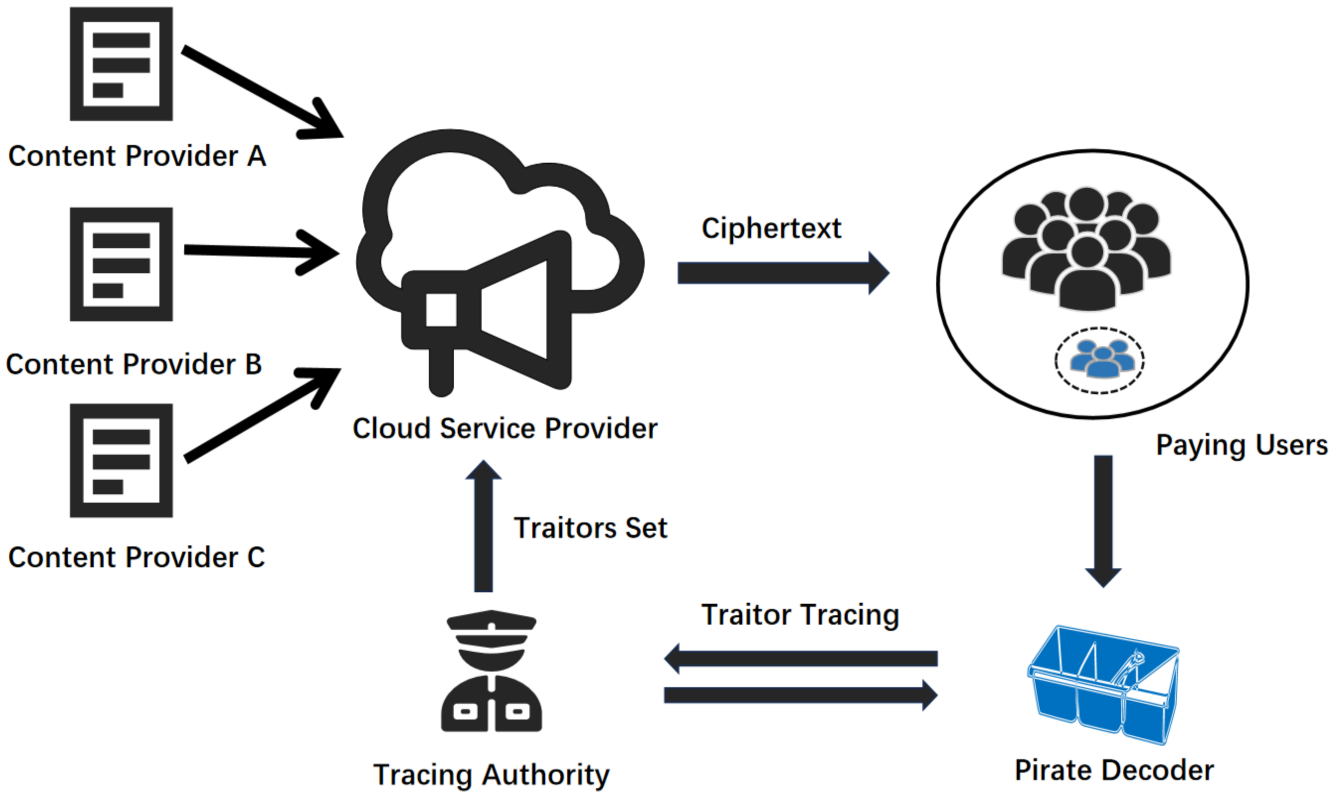

Fig. 1. System architecture of TT^+ scheme for CTV.

numbers. Since the output of the hash function is uniformly distributed, malicious encryptors cannot embed traps in random numbers. Our improved traitor tracking scheme can not only track traditional malicious users, but also prevent malicious encryptors from embedding traps in random numbers. Therefore, using our scheme in cloud TV, the cloud service provider can more securely deliver content from content providers to all paying subscribers.

Our proposed TT^+ scheme, as depicted in Figure 2, is practically applied in the could TV scenario. Suppose there is a cloud service provider who intends to encrypt paid content provided by a content provider and broadcast it to paying users. However, considering that malicious users construct pirate decoder and malicious encryptors embed trapdoors in random numbers, the cloud service provider use our TT^+ scheme to securely send content to paying users.

The implementation process of TT^+ scheme for cloud TV is as follows:

1. Setup: The cloud service provider selects a secure cryptographic hash function H and generates the public broadcast key. Once done, the cloud service provider generates their own private keys for all users, as well as a tracing secret key K for the tracing authority. The user's private key K_i can be used to decrypt the broadcast ciphertext, and the tracing authority uses the tracing secret key to execute a tracking algorithm to find the traitors.
2. Encryption: The cloud service provider uses the public broadcast key to encrypt messages sent by the content providers such that all the recipients can receive it. The cloud service provider uses the cryptographic hash function H to generate random numbers, which are used

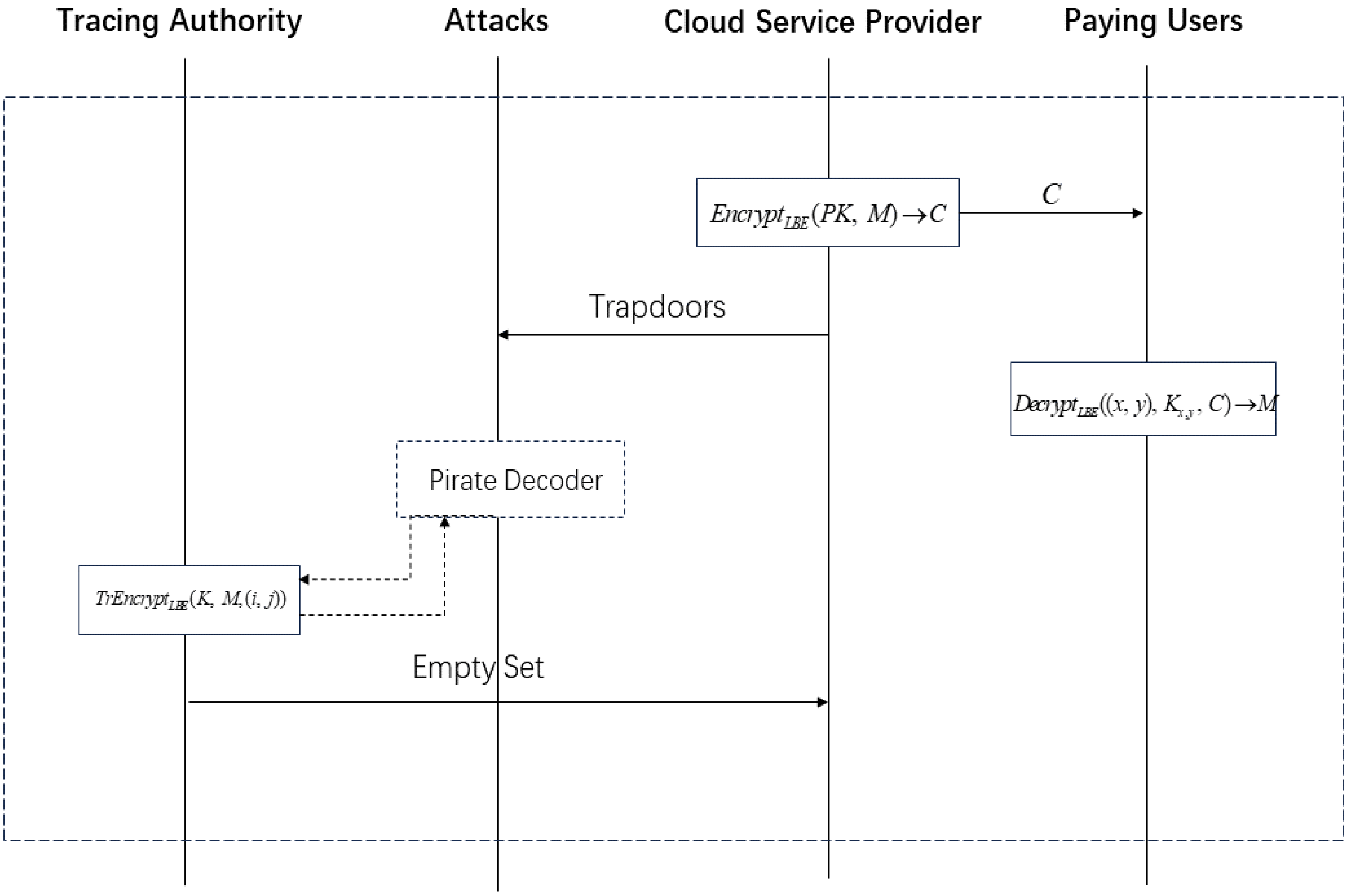
Tracing Authority
Attacks
Cloud Service Provider
Paying Users
$Encrypt_{LBE}(PK, M) \to C$
C
Trapdoors
$Decrypt_{LBE}((x, y), K_{x,y}, C) \to M$
Pirate Decoder
$TrEncrypt_{LBE}(K, M, (i, j))$
Empty Set

Fig. 2. Sequence diagram of malicious encryptors for cloud TV.

to generate row ciphertext and column ciphertexts. Once done, the cloud service provider broadcasts the ciphertexts to the paying consumers. Any paying consumer can decrypt using his private key. Of course, registered users can enforce digital rights restrictions such as do not download or play once.

3. Decryption: Paying consumers can securely decrypt the encrypted digital content and get their paid content using their own private keys.
4. TrEncryption: When a pirate decoder is found, the tracing authority can execute the tracing algorithm to interact with the pirate decoder. Finally, it outputs the index i of at least one key K_i that used to create the pirate decoder. The cloud service provider can then take legal action against the owner of this private key.

9 Conclusion and Future Work

This paper proposes a TT^+ scheme, which solves the problem of malicious encryptors embedding trapdoors in random numbers. Our scheme can be used for cloud service providers to securely

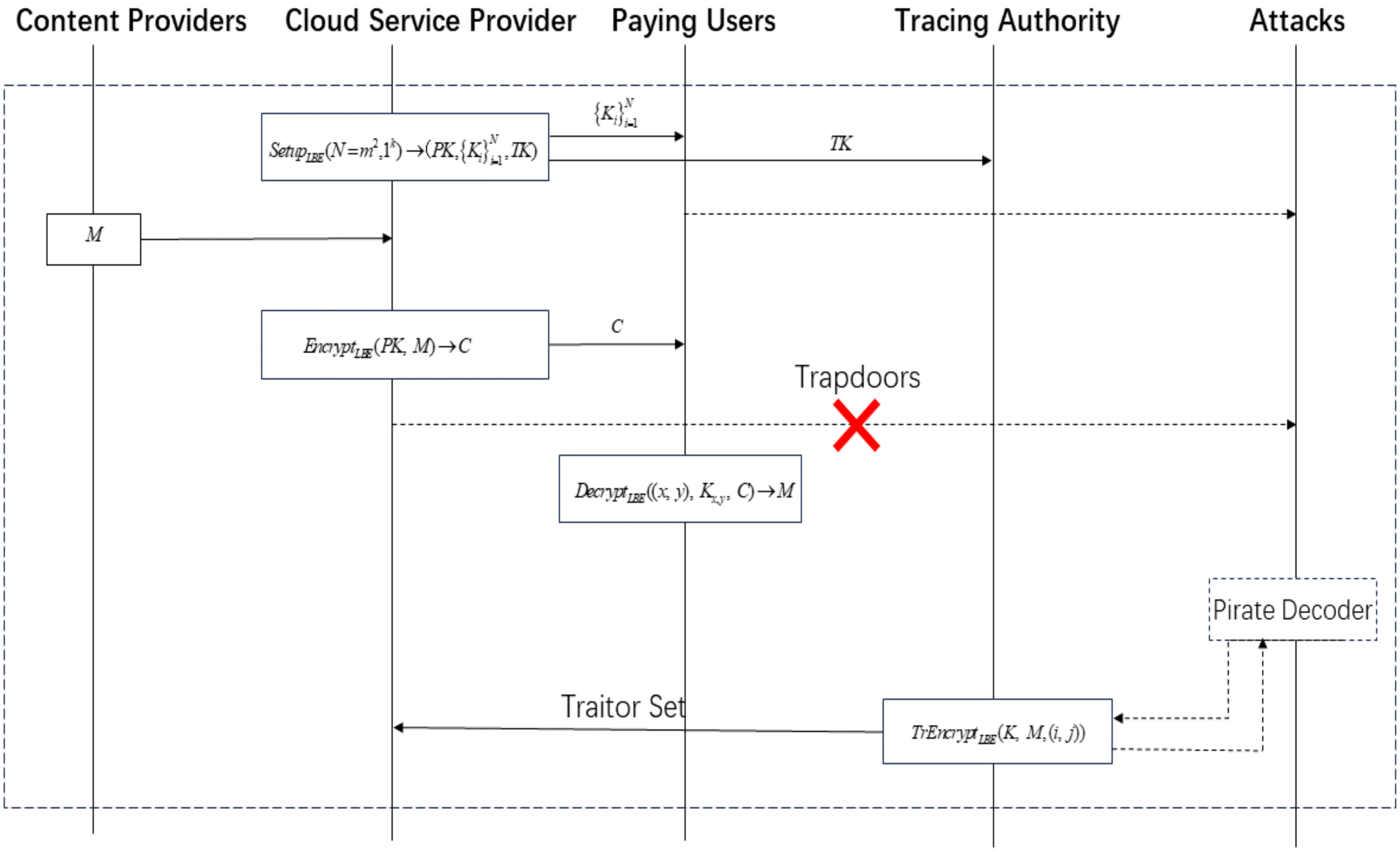
Content Providers
Cloud Service Provider
Paying Users
Tracing Authority
Attacks
$\{K_i\}_{i=1}^N$
$Setup_{LBE}(N=m^2,1^k)\rightarrow(PK,\{K_i\}_{i=1}^N,TK)$
TK
M
$Encrypt_{LBE}(PK, M)\rightarrow C$
C
Trapdoors
$Decrypt_{LBE}((x, y), K_{x,y}, C)\rightarrow M$
Pirate Decoder
Traitor Set
$TrEncrypt_{LBE}(K, M,(i, j))$

Fig. 3. Sequence diagram of TT^+ scheme for CTV.

deliver paid content to paying users in cloud TV. The random numbers used in the encryption phase of this scheme are generated by a Secure cryptographic hash function, and since the output of the hash function is uniformly distributed, this prevents malicious users from embedding trapdoors in the random numbers and giving attackers an advantage. There is a conclusion that traitor tracing and differential privacy can have directly influence on each other which has led to some computational complexity results on DP have been established on the optimal results in TT systems. Morever, also note in the literature that efficient traitor tracing schemes imply the impossibility of any differentially private data release mechanism. But this optimal result maybe no longer valid in the stronger attacker model, thus we need to carefully think the implications for differential privacy.

There are many interesting future work such as the below:

1. Differential privacy is a way to protect the privacy that may result from minor changes in data sources. Using a random response approach to make sure that the data set is always lower than a certain threshold for the output of individual records, so that a third party cannot determine whether a single record has been modified or added or deleted from the

output.It is regarded as one of the most secure methods of privacy protection. There is a close relationship between traitor tracing and differential privacy. The computational complexity results of differential privacy are established on the optimal results of traitor tracing system. Moreover, some literatures suggest that an effective traitor tracking scheme means that no differential privacy data distribution mechanism can exist. However, we introduce a new attack method and model in the traitor tracing, and in this case, the impact on differential privacy is a worthy direction to research.

2. Cryptography is critical to the security of internet communications, automobiles, and implantable medical devices. However, once large quantum computers exist, many commonly used cryptosystems will be completely broken. Post-quantum cryptography is a type of cryptography that assumes the attacker has a large quantum computer; Even in this case, post-quantum cryptography will struggle to remain safe. Because of the attack method and model of malicious encryption introduced in the traitor tracking system, the influence on post-quantum cryptography in this case is also a direction worth researching
3. Advanced access content System (AACS) is a digital rights management and content distribution standard that trys to restrict the reading and copying of a new generation of discs. AACS is a copyright protection technology and the scope of protection extends to the Internet, home network and cloud TV and so on. However, in the new attack method we introduced, malicious encryptors can embed trapdoors in random numbers, so that the attackers can ignore copyright restrictions and obtain content that should be paid for without being sanctioned, which is a big challenge for copyright protection. Therefore, the influence of this new attack method and model we introduced in traitor tracing system on AACS is also a worthy direction to research.

Acknowledgments

This work is supported by National Natural Science Foundation of China (62172436), Natural Science Foundation of Shaanxi Province (2023-JC-YB-584) and basic foundational projects of Engineering University of PAP (WJY202313).

References

1. Advanced access content system encryption. http://www.aacsla.com, 2007.
2. Michel Abdalla, Alexander W. Dent, John Malone-Lee, Gregory Neven, Duong Hieu Phan, and Nigel P. Smart. Identity-based traitor tracing. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 361–376, Beijing, China, April 16–20, 2007. Springer, Berlin, Germany.
3. Olivier Billet and Duong Hieu Phan. Traitors collaborating in public: Pirates 2.0. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 189–205, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany.
4. Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 573–592, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Berlin, Germany.
5. Dan Boneh and Brent Waters. A fully collusion resistant broadcast, trace, and revoke system. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06*, pages 211–220, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press.
6. Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 480–499, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Berlin, Germany.

7. Hervé Chabanne, Duong Hieu Phan, and David Pointcheval. Public traceability in traitor tracing schemes. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 542–558, Aarhus, Denmark, May 22–26, 2005. Springer, Berlin, Germany.
8. Benny Chor, Amos Fiat, and Moni Naor. Tracing traitors. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 257–270, Santa Barbara, CA, USA, August 21–25, 1994. Springer, Berlin, Germany.
9. Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil P. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 381–390, Bethesda, Maryland, USA, May 31 – June 2, 2009. ACM Press.
10. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49, Berkeley, CA, USA, October 26–29, 2013. IEEE Computer Society Press.
11. Sanjam Garg, Abishek Kumarasubramanian, Amit Sahai, and Brent Waters. Building efficient fully collusion-resilient traitor tracing and revocation schemes. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM CCS 10*, pages 121–130, Chicago, Illinois, USA, October 4–8, 2010. ACM Press.
12. Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 13*, pages 669–684, Berlin, Germany, November 4–8, 2013. ACM Press.
13. Aggelos Kiayias and Moti Yung. Breaking and repairing a symmetric public-key traitor tracing. In *ACM Workshop in Digital Rights Management DRM2002*, volume volume 2696 of LNCS, pages 32–50, 2002.
14. Zhen Liu, Zhenfu Cao, and Duncan S. Wong. Blackbox traceable CP-ABE: how to catch people leaking their keys by selling decryption devices on ebay. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 13*, pages 475–486, Berlin, Germany, November 4–8, 2013. ACM Press.
15. Zhen Liu, Zhenfu Cao, and Duncan S. Wong. White-box traceable ciphertext-policy attribute-based encryption supporting any monotone accesss tructures. *IEEE Transactions on Information Forensics and Security*, 8(1):76–88, 2013.
16. Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Berlin, Germany.
17. Gábor Tardos. Optimal probabilistic fingerprint codes. In *35th ACM STOC*, pages 116–125, San Diego, California, USA, June 9–11, 2003. ACM Press.
18. Jonathan Ullman. Answering $n_{2+o(1)}$ counting queries with differential privacy is hard. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 361–370, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.