A New Secure Data Deduplication Approach Supporting User Traceability

Jianfeng Wang[1], Xiaofeng Chen[1], Jin Li[2], Kamil Kluczniak[3], Mirosław Kutyłowski[3]

[1] Xidian University, Xi'an, P.R. China
[2] Guangzhou University, P.R. China
[3] Wrocław University of Technology, Poland

BWCCA 2015, Kraków

Cloud storage

Cloud storage

- the users store files in the cloud (one of primary services of the cloud)
- files encrypted by the user(s) for confidentiality and data protection

Problem

- the same data stored by many users (e.g. movie, music, . . .)
- due to encryption it might be impossible to detect the duplicates
  $\Rightarrow$ **waste of storage resources**

Goals

Requirements

1. **files must be encrypted**
2. .
3. .

4. .
5. .

   .

6. .

Goals

Requirements

1. files must be encrypted
2. **one copy per file in the cloud**
3. .
4. .
5. .

   .
6. .

Goals

Requirements

1. files must be encrypted
2. one copy per file in the cloud
3. **a user can decrypt the file iff he had the whole file at some moment**
4. .
5. .

   .
6. .

Goals

Requirements

1. files must be encrypted
2. one copy per file in the cloud
3. a user can decrypt the file iff he had the whole file at some moment
4. **no (tedious) distribution of the encryption keys**
5. .

   .
6. .

Goals

Requirements

1. files must be encrypted
2. one copy per file in the cloud
3. a user can decrypt the file iff he had the whole file at some moment
4. no (tedious) distribution of the encryption keys
5. **files signed by the uploading user in pseudonymous way**
   the signature unlinkable with the user . . .
6. .

Goals

Requirements

1. files must be encrypted
2. one copy per file in the cloud
3. a user can decrypt the file iff he had the whole file at some moment
4. no (tedious) distribution of the encryption keys
5. files signed by the uploading user in pseudonymous way
   the signature unlinkable with the user . . .
6. **. . . unless an authority enables to link his signatures due to malicious behaviour**

Convergent encryption

- encrypts/decrypts a file with a *convergent key* derived from the cryptographic hash value of the data:

$$K := \mathrm{Hash}(M), \quad C := \mathrm{Enc}_K(M)$$

- tag for referring to data:

$$T := \mathrm{Hash}(C)$$

Properties:

- anybody who knows the file knows the encryption key and the file tag
- file ciphertext easy to identify via its tag T
- three passes through M to upload M (slow!)

J. Douceur, A. Adya, W. Bolosky, P. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. ICDCS'2002

Convergent encryption
HCE1, used in Tahoe File System

- encryption

$$K := \mathrm{Hash}(M), \quad C := \mathrm{Enc}_K(M) \| \mathrm{Hash}(K)$$

- tag: $T := \mathrm{Hash}(K)$

Efficiency:

- only two passes through the data to upload (better than CE!)
- one pass to check if M already in the cloud

Convergent encryption

HCE1, used in Tahoe File System

- encryption

$$K := \mathrm{Hash}(M), \quad C := \mathrm{Enc}_K(M) \| \mathrm{Hash}(K)$$

- tag: $T := \mathrm{Hash}(K)$

Efficiency:

- only two passes through the data to upload (better than CE!)
- one pass to check if M already in the cloud

Duplicate faking attack:

- compute

$$K = \mathrm{Hash}(M), \quad C' = \mathrm{random} \| \mathrm{Hash}(K)$$

- store C' in the cloud

$\Rightarrow$

  - nobody can store M because of deduplication
  - decryption of C' yields garbage instead of M

RCE

MLE

- an abstract version of CE: message-locked encryption
- formal security model

RCE - randomized convergent encryption

- encryption:
  $K := \mathrm{Hash}(M)$
  L chosen at random
  $C_1 := \mathrm{Enc}_L(M)$, $C_2 := K \text{ XOR } L$
- tag generation:
  $T := \mathrm{Hash}(K)$

Properties:

- also more efficient than CE
- security properties due to randomization
- **vulnerable to duplicate faking attack**

M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. EUROCRYPT'13

IRCE

interactive randomized convergent encryption (IRCE)

- an honest user can check the tag by interacting with the server, and then check that the original ciphertext is stored
- an adversary may upload a ciphertext inconsistent to the file tag
- if a subsequent user attempts to upload the same file, he may claim that the ciphertext is incorrect
- the cloud server cannot check the tag against the ciphertext (the original plaintext required)
- challenges for a cloud: which user is dishonest? how to trace a malicious user?

M. Bellare and S. Keelveedhi. Interactive message-locked encryption and secure deduplication. PKC'2015

Our Solution: TrDup
outline

- each user uploading a file attaches a *traceable signature*:
  - this is a group signature – the signer remains anonymous, all one can check is that he is a registered user
  - group manager can issue a *token* that enables to trace all signatures of this user
  - traceable signatures enable erasing files uploaded by a malicious user
  - user identity hidden unless he is malicious
- proof of ownership: based on *Bloom Filters*

  a Bloom filter answers if a particular data has been inserted into it:
  - false negative answers – impossible
  - false positive answers – possible, if too many elements stored
  - the elements might come from a huge space,
    still the filter size is relatively small

Our Solution: TrDup

setup procedure

- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ – groups of a prime order p, g_1, g_2 – generators of $\mathbb{G}_1, \mathbb{G}_2$
- $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ – a bilinear mapping (that is $e(A^a, B^b) = e(A, B)^{ab}$)

setup:

$\mathbb{G}_1$:	$\mathbb{G}_2$:
m, h – random $u = h^{\xi_1^{-1}}, v = h^{\xi_2^{-1}}$ for random $\xi_1, \xi_2 < p$	ω – random $n = g_2^{\gamma}$ for random $\gamma < p$

system keys:

public keys:	private keys:
(m, n, ω, u, v, h)	ξ_1, ξ_2, γ

Wang et al.

Deduplication in Cloud

Previous work
CE
RCE

Our Solution
setup
encryption
anonymous signature
file upload

Final remarks

Issuing a “witness” for a joining user:

1. user U_i chooses x_i at random, $g_1^{x_i}$ sent to the system manager
2. the manager chooses t_i at random and uses private key γ to compute $A_i := (g^{x_i} \cdot m)^{\frac{1}{\gamma + t_i}}$
3. user U_i stores (A_i, t_i, x_i)
4. the manager stores $(i, g_1^{x_i}, A_i, t_i)$ in the user list

Wang et al.
Deduplication in Cloud
Previous work
CE
RCE
Our Solution
setup
encryption
anonymous signature
file upload
Final remarks

Issuing a “witness” for a joining user:

1. user U_i chooses x_i at random, $g_1^{x_i}$ sent to the system manager
2. the manager chooses t_i at random and uses private key γ to compute $A_i := (g^{x_i} \cdot m)^{\frac{1}{\gamma + t_i}}$
3. user U_i stores (A_i, t_i, x_i)
4. the manager stores $(i, g_1^{x_i}, A_i, t_i)$ in the user list

one can prove correctness of $(g_1^{x_i}, A_i, t_i)$ by checking

$$e(A_i, g_2^{t_i} \cdot n) \stackrel{?}{=} e(g_1^{x_i} \cdot m, g_2)$$

(indeed, $e(A_i, g_2^{t_i} \cdot n) = e((g^{x_i} \cdot m)^{\frac{1}{\gamma+t_i}}, g_2^{t_i} \cdot g_2^{\gamma}) = e(g_1^{x_i} \cdot m, g_2)^{\frac{\gamma+t_i}{t_i+\gamma}}$)

user U_i encrypts a file $\mathcal{F} = (B_0, B_1, \ldots)$

1. file encryption:
   - choose key K at random
   - compute $C_1 := \mathrm{Enc}_K(\mathcal{F})$
2. key encryption:
   - $K_{\mathcal{F}} := \mathrm{Hash}(\mathcal{F})$
   - $C_2 := K_{\mathcal{F}}\ \mathrm{XOR}\ K$
   - $T_{\mathcal{F}} := \mathrm{Hash}(K_{\mathcal{F}})$ (tag)
3. output: $(C_1, C_2, T_{\mathcal{F}})$

ownership proof generation

1. for each block i:
   - compute $E_{B_i} := \mathrm{PRNG}(\mathrm{Hash}(B_i), i)$
   - insert E_{B_i} into a Bloom filter $BF_{\mathcal{F}}$
2. store $(BF_{\mathcal{F}}, T_{\mathcal{F}})$ in index A

user U_i encrypts a file $\mathcal{F} = (B_0, B_1, \ldots)$

1. file encryption:
   - choose key K at random
   - compute $C_1 := \mathrm{Enc}_K(\mathcal{F})$
2. key encryption:
   - $K_\mathcal{F} := \mathrm{Hash}(\mathcal{F})$
   - $C_2 := K_\mathcal{F}\ \mathrm{XOR}\ K$
   - $T_\mathcal{F} := \mathrm{Hash}(K_\mathcal{F})$ (tag)
3. output: $C_\mathcal{F} = (C_1, C_2, T_\mathcal{F})$

ownership proof generation

1. for each block i:
   - compute $E_{B_i} := \mathrm{PRNG}(\mathrm{Hash}(B_i), i)$
   - insert E_{B_i} into a Bloom filter $BF_\mathcal{F}$
2. store $(BF_\mathcal{F}, T_\mathcal{F})$ in index A

TrDup

anonymous signature for a file $\mathcal{F}$

1. choose r_1, r_2, r_3 at random
2. compute $d_1 := r_1 \cdot t_i$ and $d_2 := r_2 \cdot t_i$ (blinding t_i)
3. $T_1 := u^{r_1}$, $T_2 := v^{r_2}$, $T_3 = A_i \cdot h^{r_1+r_2}$ (linear encryption of A_i)
4. $T_4 := \omega^{r_3}$, $T_5 := e(g_1, T_4)^{x_i}$
5. choose random $b_{r_1}, b_{r_2}, b_{d_1}, b_{d_2}, b_{t_i}, b_{x_i} \in \mathbb{Z}_p$
6. $B_1 := u^{b_{r_1}}$, $B_2 := v^{b_{r_2}}$,
   $B_3 := T_1^{b_{t_i}} \cdot u^{-b_{d_1}}$, $B_4 := T_2^{b_{t_i}} \cdot v^{-b_{d_2}}$,
   $B_5 := e(g_1, T_4)^{b_{x_i}}$,
   $B_6 = e(T_3, g_2)^{b_{t_i}} \cdot e(h, g_2)^{-b_{d_1}-b_{d_2}} \cdot e(h, n)^{-b_{r_1}-b_{r_2}} \cdot e(g_1, g_2)^{-b_{x_i}}$
7. compute a challenge: $c = \mathrm{Hash}(C_{\mathcal{F}}, T_1, \cdots, T_5, B_1, \cdots, B_6)$.
8. compute “Schnorr-like” signatures: $s_{r_1} = b_{r_1} + cr_1$, $s_{r_2} = b_{r_2} + cr_2$, $s_{d_1} = b_{d_1} + cd_1$, $s_{d_2} = b_{d_2} + cd_2$, $s_{x_i} = b_{x_i} + cx_i$, $s_{t_i} = b_{t_i} + ct_i$

Wang et al.

Deduplication in Cloud

Previous work
CE
RCE

Our Solution
setup
encryption
anonymous signature
file upload

Final remarks

signature of U_i for file $\mathcal{F}$:

$$(T_1, \cdots, T_5, c, s_{r_1}, s_{r_2}, s_{d_1}, s_{d_2}, s_{x_i}, s_{t_i})$$

TrDup

signature verification

1. reconstruct the B values:
   - $\tilde{B}_1 := u^{s_{r_1}} \cdot T_1^{-c}$ (proof of knowledge of r_1)
   - $\tilde{B}_2 := v^{s_{r_2}} \cdot T_2^{-c}$ (proof of knowledge of r_2)

   indeed, like for Schnorr signatures with public key $T_1 = u^{r_1}$ and secret key r_1:

   $$u^{s_{r_1}} \cdot T_1^{-c} = u^{b_{r_1} + c r_1} \cdot (u^{r_1})^{-c} = u^{b_{r_1}} = B_1$$

   - . . .

1. reconstruct the B values:
   - $\tilde{B}_1 := u^{s_{r_1}} \cdot T_1^{-c}$
   - $\tilde{B}_2 := v^{s_{r_2}} \cdot T_2^{-c}$
   - $\tilde{B}_3 := T_1^{s_{t_i}} \cdot u^{-s_{d_1}}$ (proof of knowledge of d_1)

     essentially, again a Schnorr-like signature:

     $$T_1^{s_{t_i}} \cdot u^{-s_{d_1}} = T_1^{b_{t_i}+ct_i} \cdot u^{-b_{d_1}-cd_1} = \left(T_1^{b_{t_i}} \cdot u^{-b_{d_1}}\right) \cdot T_1^{ct_i} \cdot u^{-cr_1 t_i} =$$
     $$= B_3 \cdot (u^{r_1})^{ct_i} \cdot u^{-cr_1 t_i} = B_3$$

   - $\tilde{B}_4 := T_2^{s_{t_i}} \cdot v^{-s_{d_2}}$ (proof of knowledge of d_2)
   - ...

TrDup
signature verification

1. reconstruct the B values:
   - $\tilde{B}_1 := u^{s_{r_1}} \cdot T_1^{-c}$ (proof of knowledge of r_1)
   - $\tilde{B}_2 := v^{s_{r_2}} \cdot T_2^{-c}$ (proof of knowledge of r_2)
   - $\tilde{B}_3 := T_1^{s_{t_i}} \cdot u^{-s_{d_1}}$ (proof of knowledge of d_1)
   - $\tilde{B}_4 := T_2^{s_{t_i}} \cdot v^{-s_{d_2}}$ (proof of knowledge of d_2)
   - $\tilde{B}_5 := e(g_1, T_4)^{s_{x_i}} \cdot T_5^{-c}$ (proof of knowledge of x_i but complicated, since the public key concealed)

$$e(g_1, T_4)^{s_{x_i}} \cdot T_5^{-c} = e(g_1, T_4)^{s_{x_i}} \cdot e(g_1, T_4)^{-cx_i} = e(g_1, T_4)^{b_{x_i}} = B_5$$

Wang et al.

Deduplication in Cloud

Previous work
CE
RCE

Our Solution
setup
encryption
anonymous signature
file upload

Final remarks

1. reconstruct the B values:
   - $\tilde{B}_1 := u^{s_{r_1}} \cdot T_1^{-c}$ (proof of knowledge of r_1)
   - $\tilde{B}_2 := v^{s_{r_2}} \cdot T_2^{-c}$ (proof of knowledge of r_2)
   - $\tilde{B}_3 := T_1^{s_{t_i}} \cdot u^{-s_{d_1}}$ (proof of knowledge of d_1)
   - $\tilde{B}_4 := T_2^{s_{t_i}} \cdot v^{-s_{d_2}}$ (proof of knowledge of d_2)
   - $\tilde{B}_5 := e(g_1, T_4)^{s_{x_i}} \cdot T_5^{-c}$ (proof of knowledge of x_i)
   - $\tilde{B}_6 := e(T_3, g_2)^{s_{t_i}} \cdot e(h, g_2)^{-s_{d_1}-s_{d_2}} \cdot e(h, n)^{-s_{r_1}-s_{r_2}} \cdot e(g_1, g_2)^{-s_{x_i}} \cdot e(T_3, n)^{c} \cdot e(m, g_2)^{-c}$

Correctness:

$\tilde{B}_6 =$
$e(T_3, g_2)^{s_{t_i}} \cdot e(h, g_2)^{-s_{d_1} - s_{d_2}} \cdot e(h, n)^{-s_{r_1} - s_{r_2}} \cdot e(g_1, g_2)^{-s_{x_i}} \cdot e(T_3, n)^{c} \cdot$
$e(m, g_2)^{-c} =$

Correctness:

$$\tilde{B}_6 = e(T_3, g_2)^{s_{t_i}} \cdot e(h, g_2)^{-s_{d_1} - s_{d_2}} \cdot e(h, n)^{-s_{r_1} - s_{r_2}} \cdot e(g_1, g_2)^{-s_{x_i}} \cdot e(T_3, n)^{c} \cdot e(m, g_2)^{-c} =$$

$$e(T_3, g_2)^{b_{t_i} + ct_i} \cdot e(h, g_2)^{-b_{d_1} - cd_1 - b_{d_2} + cd_2} \cdot e(h, n)^{-b_{r_1} - cr_1 - b_{r_2} - cr_2} \cdot e(g_1, g_2)^{-b_{x_i} - cx_i} \cdot e(T_3, n)^{c} \cdot e(m, g_2)^{-c} =$$

Correctness:

$$\tilde{B}_6 = e(T_3, g_2)^{s_{t_i}} \cdot e(h, g_2)^{-s_{d_1} - s_{d_2}} \cdot e(h, n)^{-s_{r_1} - s_{r_2}} \cdot e(g_1, g_2)^{-s_{x_i}} \cdot e(T_3, n)^c \cdot e(m, g_2)^{-c} =$$

$$e(T_3, g_2)^{b_{t_i} + ct_i} \cdot e(h, g_2)^{-b_{d_1} - cd_1 - b_{d_2} + cd_2} \cdot e(h, n)^{-b_{r_1} - cr_1 - b_{r_2} - cr_2} \cdot e(g_1, g_2)^{-b_{x_i} - cx_i} \cdot e(T_3, n)^c \cdot e(m, g_2)^{-c} =$$

$$\left(e(T_3, g_2)^{b_{t_i}} \cdot e(h, g_2)^{-b_{d_1} - b_{d_2}} \cdot e(h, n)^{-b_{r_1} - b_{r_2}} \cdot e(g_1, g_2)^{-b_{x_i}}\right) \cdot e(T_3, g_2)^{ct_i} \cdot e(h, g_2)^{-cd_1 - cd_2} \cdot e(h, n)^{-cr_1 - cr_2} \cdot e(g_1, g_2)^{-cx_i} \cdot e(T_3, n)^c \cdot e(m, g_2)^{-c} =$$

Wang et al.

Deduplication in Cloud

Previous work

CE

RCE

Our Solution

setup

encryption

anonymous signature

file upload

Final remarks

Correctness:

$$\tilde{B}_6 = e(T_3, g_2)^{s_{t_i}} \cdot e(h, g_2)^{-s_{d_1} - s_{d_2}} \cdot e(h, n)^{-s_{r_1} - s_{r_2}} \cdot e(g_1, g_2)^{-s_{x_i}} \cdot e(T_3, n)^{c} \cdot e(m, g_2)^{-c} =$$

$$e(T_3, g_2)^{b_{t_i} + ct_i} \cdot e(h, g_2)^{-b_{d_1} - cd_1 - b_{d_2} + cd_2} \cdot e(h, n)^{-b_{r_1} - cr_1 - b_{r_2} - cr_2} \cdot e(g_1, g_2)^{-b_{x_i} - cx_i} \cdot e(T_3, n)^{c} \cdot e(m, g_2)^{-c} =$$

$$\left(e(T_3, g_2)^{b_{t_i}} \cdot e(h, g_2)^{-b_{d_1} - b_{d_2}} \cdot e(h, n)^{-b_{r_1} - b_{r_2}} \cdot e(g_1, g_2)^{-b_{x_i}}\right) \cdot e(T_3, g_2)^{ct_i} \cdot e(h, g_2)^{-cd_1 - cd_2} \cdot e(h, n)^{-cr_1 - cr_2} \cdot e(g_1, g_2)^{-cx_i} \cdot e(T_3, n)^{c} \cdot e(m, g_2)^{-c} =$$

$$B_6 \cdot \left(e(T_3, g_2)^{t_i} \cdot e(h, g_2)^{-d_1 - d_2} \cdot e(h, n)^{-r_1 - r_2} \cdot e(g_1, g_2)^{-x_i} \cdot e(T_3, n) \cdot e(m, g_2)^{-1}\right)^{c}$$

Wang et al.
Deduplication in Cloud
Previous work
CE
RCE
Our Solution
setup
encryption
anonymous signature
file upload
Final remarks

Correctness: it suffices to show that the below expression equals 1:

$$e(T_3, g_2)^{t_i} \cdot e(h, g_2)^{-d_1-d_2} \cdot e(h, n)^{-r_1-r_2} \cdot e(g_1, g_2)^{-x_i} \cdot e(T_3, n) \cdot e(m, g_2)^{-1} =$$

$$e(A_i \cdot h^{r_1+r_2}, g_2)^{t_i} \cdot e(h, g_2)^{t_i(-r_1-r_2)} \cdot e(h, n)^{-r_1-r_2} \cdot e(g_1, g_2)^{-x_i} \cdot e(A_i \cdot h^{r_1+r_2}, n) \cdot e(m, g_2)^{-1} =$$

.

.

Correctness: it suffices to show that the below expression equals 1:

$$e(T_3, g_2)^{t_i} \cdot e(h, g_2)^{-d_1-d_2} \cdot e(h, n)^{-r_1-r_2} \cdot e(g_1, g_2)^{-x_i} \cdot e(T_3, n) \cdot e(m, g_2)^{-1} =$$

$$e(A_i \cdot h^{r_1+r_2}, g_2)^{t_i} \cdot e(h, g_2)^{t_i(-r_1-r_2)} \cdot e(h, n)^{-r_1-r_2} \cdot e(g_1, g_2)^{-x_i} \cdot e(A_i \cdot h^{r_1+r_2}, n) \cdot e(m, g_2)^{-1} =$$

$$e(A_i, g_2)^{t_i} \cdot e(g_1, g_2)^{-x_i} \cdot e(A_i, n) \cdot e(m, g_2)^{-1} =$$

.

TrDup

signature verification

Correctness: it suffices to show that the below expression equals 1:

$$e(T_3, g_2)^{t_i} \cdot e(h, g_2)^{-d_1-d_2} \cdot e(h, n)^{-r_1-r_2} \cdot e(g_1, g_2)^{-x_i} \cdot e(T_3, n) \cdot e(m, g_2)^{-1} =$$

$$e(A_i \cdot h^{r_1+r_2}, g_2)^{t_i} \cdot e(h, g_2)^{t_i(-r_1-r_2)} \cdot e(h, n)^{-r_1-r_2} \cdot e(g_1, g_2)^{-x_i} \cdot e(A_i \cdot h^{r_1+r_2}, n) \cdot e(m, g_2)^{-1} =$$

$$e(A_i, g_2)^{t_i} \cdot e(g_1, g_2)^{-x_i} \cdot e(A_i, n) \cdot e(m, g_2)^{-1} =$$

$$e(A_i, g_2^{t_i} \cdot n) \cdot e(g_1^{x_i} \cdot m, g_2)^{-1} = 1$$

(the last equation holds since A_i, t_i, x_i satisfy $e(A_i, g_2^{t_i} \cdot n) = e(g_1^{x_i} \cdot m, g_2)$)

TrDup
signature verification

1. reconstruct the B values:
   - $\tilde{B}_1 = u^{s_{r_1}} \cdot T_1^{-c}$
   - $\tilde{B}_2 = v^{s_{r_2}} \cdot T_2^{-c}$
   - $\tilde{B}_3 = T_1^{s_{t_i}} \cdot u^{-s_{d_1}}$
   - $\tilde{B}_4 = T_2^{s_{t_i}} \cdot v^{-s_{d_2}}$
   - $\tilde{B}_5 = e(g_1, T_4)^{s_{x_i}} \cdot T_5^{-c}$
   - $\tilde{B}_6 = e(T_3, g_2)^{s_{t_i}} \cdot e(h, g_2)^{-s_{d_1} - s_{d_2}} \cdot e(h, n)^{-s_{r_1} - s_{r_2}} \cdot e(g_1, g_2)^{-s_{x_i}} \cdot e(T_3, n)^{c} \cdot e(m, g_2)^{-c}$
2. test: $c \stackrel{?}{=} H(C_{\mathcal{F}}, T_1, \cdots, T_5, \tilde{B}_1, \cdots, \tilde{B}_6)$

Upload request of a file with tag $T_{\mathcal{F}}$

- if no file with tag $T_{\mathcal{F}}$ on the cloud server and the signature is valid
  $\Rightarrow$
  the cloud server stores the ciphertext of file $T_{\mathcal{F}}$
- if a duplicate found
  $\Rightarrow$
  the user proves that he holds the file

TrDup

proof of possession details

Wang et al.

Deduplication in Cloud

Previous work
CE
RCE

Our Solution
setup
encryption
anonymous signature
file upload

Final remarks

1. **cloud server:** choose at random q blocks: $B_{j_1}, \ldots, B_{j_q}$
2. **user:** for $i = 1, \ldots, q$, compute the token $T_j := \mathrm{Hash}(B_{j_i})$
3. **cloud server:**
   1. $E_{B_{j_i}} := \mathrm{PRNG}(T_{B_{j_i}}, j_i)$ for $i = 1, \ldots, q$
   2. if some $E_{B_{j_i}}$ does not belong to the Bloom filter $BF_{\mathcal{F}}$, then abort
   3. the link to the encrypted file $\mathcal{F}$ is given assigned to the user
4. **cloud server:** $h := \mathrm{Hash}(C_1)$, send h and C_2 to the user
5. **user:**
   1. $K' := C_2 \oplus K_{\mathcal{F}}$ (reconstruction of the file encryption key)
   2. check $\mathrm{Hash}(Enc_{K'}(\mathcal{F})) \stackrel{?}{=} h$ (check if the same files are stored)
   3. if not, then investigation started to reveal the user who uploaded the (invalid) file

processing of a tracing request by the group manager:

1. **checking correctness of the file $\mathcal{F}$** by the group manager:
   1. (re)compute the tag $T' := \mathrm{Hash}(\mathrm{Hash}(\mathcal{F}))$ and request the file ciphertext $C_{\mathcal{F}} = (C_1, C_2, T_{\mathcal{F}})$ from the cloud
   2. recover the file encryption key $K' := C_2 \oplus H(\mathcal{F})$, decrypt $\mathcal{F}' := \mathrm{Dec}_{K'}(C_1)$
   3. check $\mathrm{Hash}(\mathcal{F}') \stackrel{?}{=} \mathrm{Hash}(\mathcal{F})$

processing of a tracing request by the group manager:

1. **checking correctness of the file $\mathcal{F}$** by the group manager:
   1. (re)compute the tag $T' := \mathrm{Hash}(\mathrm{Hash}(\mathcal{F}))$ and request the file ciphertext $C_{\mathcal{F}} = (C_1, C_2, T_{\mathcal{F}})$ from the cloud
   2. recover the file encryption key $K' := C_2 \oplus H(\mathcal{F})$, decrypt $\mathcal{F}' := \mathrm{Dec}_{K'}(C_1)$
   3. check $\mathrm{Hash}(\mathcal{F}') \stackrel{?}{=} \mathrm{Hash}(\mathcal{F})$
2. **deanonymization if the ciphertext invalid** by the group manager:
   1. get $\tilde{A} := T_3/(T_1^{\xi_1} \cdot T_2^{\xi_2})$ using the master private keys ξ_1, ξ_2

      note: $T_3/(T_1^{\xi_1} \cdot T_2^{\xi_2}) = (A_i \cdot h^{r_1+r_2})/[(u^{r_1})^{\xi_1} \cdot (u^{r_2})^{\xi_2}] = A_i$
   2. find $A_i = \tilde{A}$ in the user list L storing records $(i, g_1^{x_i}, A_i, t_i)$

processing of a tracing request by the group manager:

1. **checking correctness of the file $\mathcal{F}$** by the group manager:
   1. (re)compute the tag $T' := \mathrm{Hash}(\mathrm{Hash}(\mathcal{F}))$ and request the file ciphertext $C_{\mathcal{F}} = (C_1, C_2, T_{\mathcal{F}})$ from the cloud
   2. recover the file encryption key $K' := C_2 \oplus H(\mathcal{F})$, decrypt $\mathcal{F}' := \mathrm{Dec}_{K'}(C_1)$
   3. check $\mathrm{Hash}(\mathcal{F}') \stackrel{?}{=} \mathrm{Hash}(\mathcal{F})$
2. **deanonymization if the ciphertext invalid** by the group manager:
   1. get $\tilde{A} := T_3/(T_1^{\xi_1} \cdot T_2^{\xi_2})$ using the master private keys ξ_1, ξ_2

      note: $T_3/(T_1^{\xi_1} \cdot T_2^{\xi_2}) = (A_i \cdot h^{r_1+r_2})/[(u^{r_1})^{\xi_1} \cdot (u^{r_2})^{\xi_2}] = A_i$
   2. find $A_i = \tilde{A}$ in the user list L storing records $(i, g_1^{x_i}, A_i, t_i)$
3. **tracing malicious user** by tracing agents:
   1. get $g_1^{x_k}$ from the group manager
   2. detect a signature from the user U_k: check $e(g_1^{x_k}, T_4) \stackrel{?}{=} T_5$

      note: $e(g_1^{x_k}, T_4) = e(g_1, T_4)^{x_k} \stackrel{def}{=} T_5$

Conclusions

- most overhead related to Bloom Filter
- traceable signature: complicated. Problem for a security proof (boring) and transparency for the users rather than computational problem
- registration needed – PKI is always problematic in a world wide environment

Thanks for your attention!

Contact data

1. `Miroslaw.Kutylowski@pwr.edu.pl`
2. `http://kutylowski.im.pwr.edu.pl`
3. `http://ki.pwr.edu.pl`

supported by National Natural Science Foundation of China (No. 61272455), China 111 Project (No. B08038), National High Tech. R&D Program (863 Program) of China (No. 2015AA016007), Doctoral Fund of Min. of Education of China (No. 20130203110004), Program for New Century Excellent Talents in University (No. NCET-13-0946), the Fundamental Research Funds for the Central Universities (No. BDY151402) and Fujian Provincial Key Laboratory of Network Security and Cryptology Research Fund (Fujian Normal University) (No. 15009), Wrocław University of Technology Poland-China cooperation project