#

Publicly Verifiable Attribute-Based Data Recoverability Proof Scheme Postprint

Authors: Ren Yan, Tang Chunming

Date: 2018-12-13T00:00:00+00:00

Abstract

Proof of retrievability schemes can effectively solve the problem of verifying data integrity when users store data on untrusted servers. Addressing the issue that most existing schemes employ identity-based cryptography, we design an attribute-based proof of retrievability scheme using the more intuitive and flexible attribute-based cryptography. We present the relevant definitions, security model, and concrete construction of the scheme, and simultaneously prove its correctness and security.

Full Text

Preamble

Vol. 37 No. 2
Application Research of Computers
ChinaXiv Partner Journal

Accepted Paper

Attribute-Based Proof of Retrievability with Public Verifiability

Ren Yan[1],[2], *Tang Chunming*[2]
(1. School of Mathematics & Information Technology, Yuncheng University, Yuncheng Shanxi 044000, China;
2. Key Laboratory of Information Security Technology, School of Mathematics and Information Sciences, Guangzhou University, Guangzhou 510006, China)

Abstract: When users store data on dishonest servers, proof-of-retrievability (POR) systems are proposed to verify data integrity. Most existing solutions employ identity-based cryptosystems. To address this limitation, this paper adopts a more intuitive and flexible attribute-based cryptosystem to design an attribute-based data recoverability proof scheme. We present the scheme' s definitions, security model, and concrete construction, and prove its correctness and security.

Key words: cloud computing; attribute-based cryptography; proof-of-retrievability; public verifiability

0 Introduction

In outsourced data storage services, enabling data owners to effectively and securely verify that storage servers correctly store their data is critically important. To address this problem, Juels et al. [1] first formally proposed the concept and scheme of Proof of Retrievability (POR). In POR schemes, storage servers must prove to users that they have correctly stored the data and convince users that they can recover previously stored files. In [1], the scheme detects server data modifications by embedding a sentinel block among invariable file blocks, with communication cost linear in the number of elements per file block. Since then, many scholars have studied POR schemes. Recent research achievements include: constant-communication POR schemes in [2,3]; Zhu et al. [4] proposed formal definitions of interactive provable retrievability under the standard model of interactive proof systems, and presented a practical zero-knowledge proof of retrievability. This scheme was proven to have completeness, soundness, and zero-knowledge properties under the Diffie-Hellman assumption. The proof implements a polynomial-time knowledge extractor, and the protocol achieves commitment, challenge, and response processes by transmitting only fixed-size data, minimizing network communication. Therefore, this scheme can be used for public remote verification of large files in large-scale distributed storage systems. Reference [5] proposed two efficient dynamic POR schemes —one for private verification and one for public verification. References [6–9] respectively proposed several dynamic publicly verifiable POR schemes.

Most existing schemes employ identity-based cryptography. This paper considers applying attribute-based cryptography to POR. Compared with identity-based cryptography, attribute-based cryptography is more intuitive. For example, in signature schemes, when someone signs a message using an identity-based signature, the verifier can confirm the signature indeed comes from that person, but learns nothing about that person' s permissions or social functions. In attribute-based signatures, however, the verifier can verify whether the signature comes from a holder of the appropriate attributes, thereby understanding the signer' s permissions and functions while maintaining anonymity of the signer' s identity. Using real-world seals as an analogy: identity-based signatures are like personal seals, while attribute-based signatures are like official seals. A personal seal only identifies the responsible individual, whereas an official seal reveals the issuing organization or attributes. In practice, official seals are clearly more credible than personal signature seals.

This paper first proposes the definition and security model for attribute-based publicly verifiable proof-of-retrievability schemes, implements a concrete construction, and provides proofs of correctness and security. In our scheme, users

only know the file owner' s attributes without learning any identity information about the file owner.

1.1 Bilinear Maps

Let G_1 and G_2 be two cyclic multiplicative groups of prime order q. Let $e : G_1 \times G_2 \to G_T$ be a bilinear map. Assuming the discrete logarithm problem (DLP) is hard in both G_1 and G_2, the bilinear map satisfies the following properties:

a) **Bilinearity:** For all $P \in G_1$, $Q \in G_2$, and $a, b \in Z_q$, we have $e(P^a, Q^b) = e(P, Q)^{ab}$.

b) **Non-degeneracy:** There exist $P \in G_1$ and $Q \in G_2$ such that $e(P, Q) \neq 1$.

c) **Computability:** There exists an efficient algorithm to compute $e(P, Q)$ for any $P \in G_1$, $Q \in G_2$.

1.2 Lagrange Interpolation Theorem

Let $f(x)$ be a polynomial of degree n over Z_q. Given $n + 1$ distinct points $(x_i, f(x_i))$, the polynomial $f(x)$ can be uniquely reconstructed. For any $x \in Z_q$, we can define the Lagrange coefficient $\Delta_{i,S}$ where S is a set of elements from Z_q:

$$\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x - j}{i - j}$$

1.3 Hardness Assumptions

This scheme relies on the following hard problems:

a) **Discrete Logarithm Problem (DLP):** Let G be an additive cyclic group of prime order q. Given $P, Q \in G$ where $Q = aP$ for some unknown $a \in Z_q$, no polynomial-time algorithm can compute a with non-negligible advantage.

b) **Bilinear Pairing Inversion Problem:** Let $e : G \times G \to G_T$ be a bilinear map where G is an additive cyclic group of prime order q. Given $P \in G$ and $e(P, Q) \in G_T$, no polynomial-time algorithm can compute $Q \in G$ with non-negligible advantage.

c) **Computational Diffie-Hellman Problem (CDH):** Let G be a multiplicative cyclic group of prime order q. Given $g, g^a, g^b \in G$ for random $a, b \in Z_q$, no polynomial-time algorithm can compute g^{ab} with non-negligible advantage.

---

2 Definition and Security Model for Attribute-Based Publicly Verifiable POR

Generally, an attribute-based publicly verifiable proof-of-retrievability scheme involves three parties: the data owner, the client, and the cloud service provider. The data owner collects data and, to improve robustness, encodes it using error-correcting codes before storing it in the cloud. Clients satisfying the data owner' s specified attributes can access the encoded data and verify its integrity. To verify integrity, the client generates a challenge message and sends it to the cloud. The cloud then computes a proof of response for the selected file blocks. Upon receiving the proof, the client can verify data integrity through a verification algorithm.

2.1 Formal Definition

Let Γ be the set of possible attributes. An assertion on Γ is essentially a Boolean function of attributes. When an attribute set ω satisfies an assertion Γ, we say ω fulfills Γ. The main algorithms are:

- **Setup:** Generates the master public key and master secret key.
- **KeyGen:** Generates secret keys for attribute sets.
- **Outsource:** Two-phase process: (1) Encoding—takes the user' s secret key and file M as input, outputs an encoded file of size $|M| \cdot \rho$; (2) Authentication tag computation—takes the encoded file as input, outputs tags.
- **Proof:** When challenged, the server generates a proof for the user.
- **Verify:** The user verifies whether the file is correctly stored.

Formally, an attribute-based publicly verifiable proof-of-retrievability scheme consists of five algorithms:

Definition 1 An attribute-based publicly verifiable proof-of-retrievability scheme comprises:

a) **Setup(1^λ):** Takes security parameter λ as input, outputs system parameters $params$, master public key pk, and master secret key msk.

b) **KeyGen(msk, ω):** Takes master secret key msk and attribute set ω as input, outputs secret key sk_ω for the attribute set.

c) **Outsource(sk_ω, M):** Takes secret key sk_ω and file M as input, outputs encoded file M' and authentication tags σ.

d) **Proof(**$params$**,** Γ**,** ω**,** M'**,** σ**):** Takes system parameters, assertion Γ, attribute set ω satisfying Γ, and encoded file M' as input, outputs proof response prf.

e) **Verify(**$params$**,** prf**):** Takes system parameters and proof response prf as input, outputs either **accept** or **reject**.

2.2 Security Model

This section describes the security model for attribute-based publicly verifiable POR schemes. Following [11-13], we consider the storage service provider as untrusted and potentially malicious. The scheme must satisfy both **correctness** and **soundness**.

Definition 2 (Correctness). A scheme is correct if for any valid proof generated by the algorithms (KeyGen, Outsource, Proof), the Verify algorithm always outputs **accept**.

Definition 3 (Soundness). If any malicious cloud service provider can generate a proof that passes verification (i.e., the verification algorithm believes it correctly stored file M), then it must possess the required file. Following the security model for soundness proposed in [10-12], we define the following game:

a) **Setup:** The challenger runs the Setup algorithm to obtain key pairs (pk, sk) and sends pk to the adversary.

b) **Outsource:** The adversary selects a file M and sends it to the challenger. The challenger runs the Outsource algorithm and responds with the encoded file.

c) **Proof:**

- (a) The challenger randomly generates a challenge message and sends it to the adversary.
- (b) Since the adversary may have lost or modified part of file M, it first randomly generates a file M'.
- (c) The adversary runs arbitrary algorithms to generate a proof prf and sends prf to the challenger.

d) **Verify:** The challenger runs the verification algorithm to check prf. The adversary wins the game if and only if it can generate a proof prf for file M that causes the challenger to output **accept**.

A scheme is sound if any probabilistic polynomial-time adversary wins the above game with only negligible probability.

---

3 Concrete Construction of Attribute-Based Publicly Verifiable POR

Assume the attribute set Γ contains n attributes, where each element corresponds to a unique integer in Z_q. In this scheme, the file M can be of arbitrary bit length. We now describe the attribute-based publicly verifiable POR scheme that supports all assertions Γ. Specifically, for threshold value d, we have concrete construction as follows:

1) Setup

The Setup phase completes the following steps:

a) Define the attribute set Γ. For simplicity, let $\Gamma = \{1, 2, ..., n\}$ and we can take the first l elements as this set, i.e., $\Gamma = \{1, 2, ..., l\}$.

b) Let G be a multiplicative cyclic group of prime order p. Randomly select a generator $g \in G$.

c) Randomly select a vector $U = (u_1, u_2, ..., u_n)$ of length n where each $u_i \in G$.

d) Randomly select $x \in Z_p^*$ and compute $X = g^x$.

The public parameters are $params = (g, X, d, U)$. The master secret key is $msk = x$.

2) Key Generation

For a user' s attribute set $\omega \subseteq \Gamma$, generate the secret key as follows:

a) Select a $d-1$ degree polynomial $q(x)$ such that $q(0) = x$.

b) For each $i \in \omega$, compute $d_i = g^{q(i)}$.

c) Output $sk_\omega = \{d_i\}_{i \in \omega}$ as the secret key.

3) Outsourcing Storage

a) **Encoding:** Given a data file M, apply error-correcting codes to obtain the encoded file M'. Split M' into n blocks, where each block $\mu_i \in \{0, 1\}^t$.

b) **Authentication Tag Computation:** For each data block, compute the authentication tag as follows:

- Randomly select a file name $name$ and a value $s \in Z_p^*$, and publish $name$.
- For each block $i \in \{1, 2, ..., n\}$, compute:

$$\sigma_i = \left(H(name) \cdot \prod_{j \in S} u_j^{\Delta_{j,S}(0)} \right)^{\frac{1}{s+\mu_i}}$$

where S is a subset of ω with $|S| = d$.

c) Output the encoded file $M' = \{\mu_i\}_{i=1}^n$ and tags $\{\sigma_i\}_{i=1}^n$.

d) Outsource $(M', \{\sigma_i\})$ to the cloud server for storage.

4) Verify Challenge

To verify that the server correctly stores the file, the client randomly selects a subset $S \subseteq \omega$ with $|S| = d$ and a random value $r \in Z_p^*$, then sends the challenge $chal = (S, r)$ to the server.

5) Proof Generation (ProofGen)

Upon receiving the challenge, the server computes:

$$\sigma = \prod_{i \in S} \sigma_i^{\Delta_{i,S}(0)}, \quad \psi = \prod_{i \in S} \mu_i^{\Delta_{i,S}(0)}$$

The server sends the proof $prf = (\sigma, \psi)$ to the client.

6) Verification (Verify)

After receiving proof prf, the client first computes:

$$\eta = H(name) \cdot \prod_{i \in S} u_i^{\Delta_{i,S}(0)}$$

Then verifies the following equation to check whether the server correctly stored the file without tampering:

$$e(\sigma, \eta) = e(g^r, X) \cdot e(\psi, g)$$

4.1 Correctness Analysis

Theorem 1. The verification process of the scheme is correct.

Proof. For a valid proof generated from correctly stored data, we have:

$$e(\sigma, \eta) = e\left(\prod_{i \in S} \sigma_i^{\Delta_{i,S}(0)}, H(name) \cdot \prod_{i \in S} u_i^{\Delta_{i,S}(0)}\right)$$

Expanding using the tag computation:

$$= e\left(\prod_{i \in S}\left(H(name) \cdot \prod_{j \in S} u_j^{\Delta_{j,S}(0)}\right)^{\frac{\Delta_{i,S}(0)}{s+\mu_i}}, H(name) \cdot \prod_{i \in S} u_i^{\Delta_{i,S}(0)}\right)$$

By the properties of bilinear maps and Lagrange interpolation, this simplifies to:

$$= e(g^r, X) \cdot e(\psi, g)$$

Thus the verification equation holds, proving correctness.

4.2 Security Analysis

Theorem 2. The proof prf in our scheme satisfies soundness under the CBDH assumption; that is, it is unforgeable.

Proof. Assume a probabilistic polynomial-time adversary $\mathcal{A}$ can win the soundness game with non-negligible advantage. We construct an algorithm $\mathcal{B}$ that solves the CDH problem. Given (g, g^a, g^b), algorithm $\mathcal{B}$ proceeds as follows:

1) Setup Simulation: $\mathcal{B}$ randomly selects $x \in Z_p^*$ and sets $X = g^x$. It outputs the challenge assertion Γ with threshold d on attribute set ω.

2) Key Generation Simulation: For private key queries on attribute set ω', $\mathcal{B}$: - Randomly selects $s \in Z_p^*$. - Defines three sets $\Gamma_0, \Gamma_1, \Gamma_2$ satisfying $|\omega' \cap \Gamma| \geq d$. - For each $i \in \omega'$, computes $d_i = g^{\tau_i}$ where τ_i are properly simulated.

3) Outsourcing Storage Simulation: For file M: - Randomly selects file name $name$ and value $s \in Z_p^*$. - Computes tags σ_i using the simulated keys.

4) Challenge Simulation: The client randomly selects subset $S \subseteq \omega$ with $|S| = d$ and random $r \in Z_p^*$, sending $chal = (S, r)$ to the server.

5) Proof Generation Simulation: The adversary computes $\sigma' \neq \sigma$ and $\psi' = \psi$.

6) Verification: The challenger computes:

$$\frac{e(\sigma, \eta)}{e(\sigma', \eta)} = \frac{e(g^r, X) \cdot e(\psi, g)}{e(g^r, X) \cdot e(\psi', g)}$$

Since $\psi = \psi'$ but $\sigma \neq \sigma'$, we get $e(\sigma/\sigma', \eta) = 1$. This implies the adversary can compute g^{ab}, solving the CBDH problem. This contradicts the CBDH assumption, proving the scheme' s soundness.

Three cases are analyzed: 1. $\sigma \neq \sigma', \psi = \psi'$: The adversary can compute g^{ab}. 2. $\sigma = \sigma', \psi \neq \psi'$: Similarly leads to solving CBDH. 3. $\sigma \neq \sigma', \psi \neq \psi'$: Also reduces to CBDH.

In all cases, the adversary' s ability to forge a proof would break the CBDH assumption. Therefore, the scheme is secure.

---

5 Conclusion

This paper is the first to apply the concept of attribute-based signatures to proofs of retrievability. We propose definitions and a security model for attribute-based publicly verifiable proof-of-retrievability schemes, implement a concrete construction, and provide proofs of correctness and security. In our scheme, users only know the file owner' s attributes without learning any identity information about the file owner.

References

[1] Juels A, Jr Kaliski B S. PORs: proofs of retrievability for large files [C]//Proc of the 14th ACM Conference on Computer and Communications Security. New York: ACM Press, 2007: 584-597.

[2] Xu Jia, Chang E C. Towards efficient provable data possession [J]. IACR Cryptology ePrint Archive, 2011, 2011: 574.

[3] Yuan Jiawei, Yu Shucheng. Proofs of retrievability with public verifiability and constant communication cost in cloud [C]//Proc of International Workshop on Security in Cloud Computing. 2013: 19-26.

[4] Zhu Yan, Wang Huaixi, Hu Zexing, et al. Zero-knowledge proofs of retrievability [J]. Science China Information Sciences, 2011, 54(8): 1604-1615.

[5] Shi E, Stefanov E, Papamanthou C. Practical dynamic proofs of retrievability [C]//Proc of ACM Conference on Computer and Communications Security. New York: ACM Press, 2013: 325-336.

[6] Beng T C, Hijazi M H A, Lim Y, et al. A survey on Proof of Retrievability for cloud data integrity and availability: cloud storage state-of-the-art, issues, solutions and future trends [J]. Journal of Network and Computer Applications, 2018, 110: 75-86.

[7] Ren Zhengwei, Wang Lina, Wang Qian, et al. Dynamic proofs of retrievability for coded cloud storage systems [J]. IEEE Trans on Services Computing, 2018, 11(4): 685-698.

[8] Fu Anmin, Li Yuhan, Yu Shui, et al. DIPOR: an IDA-based dynamic proof of retrievability scheme for cloud storage systems [J]. Journal of Network and Computer Applications, 2018, 104: 97-106.

[9] Sengupta B, Ruj S. Efficient proofs of retrievability with public verifiability for dynamic cloud storage [J]. IEEE Trans on Cloud Computing, 2017.

[10] Cash D, Küpçü A, Wichs D. Dynamic proofs of retrievability via oblivious RAM [J]. Journal of Cryptology, 2017, 30(1): 22-57.

[11] Juels A, Kaliski Jr B S. PORs: Proofs of retrievability for large files

[C]//Proc of the 14th ACM Conference on Computer and Communications Security. ACM, 2007: 584-597.

[12] Shacham H, Waters B. Compact proofs of retrievability [C]//Advances in Cryptology-Asiacrypt. Berlin: Springer, 2008: 90-107.

Note: Figure translations are in progress. See original paper for figures.