Privacy-preserving security solution for cloud services

L. Malina*, J. Hajny, P. Dzurenda and V. Zeman

Department od Telecommunications
Brno University of Technology
Brno, Czech Republic
*malina@feec.vutbr.cz

ABSTRACT
We propose a novel privacy-preserving security solution for cloud services. Our solution is based on an efficient non-bilinear group signature scheme providing the anonymous access to cloud services and shared storage servers. The novel solution offers anonymous authenticationfor registered users. Thus, users' personal attributes (age, valid registration, successful payment) can be proven without revealing users' identity, and users can use cloud services without any threat of profiling their behavior. However, if a user breaks provider's rules, his access right is revoked.Our solution provides anonymous access, unlinkability and the confidentiality of transmitted data. We implement our solution as a proof of concept applicationand present the experimental results. Further, we analyzecurrent privacy preserving solutions for cloud services and group signature schemes as basic parts of privacy enhancing solutions in cloud services. We compare the performance of our solution with the related solutionsand schemes.

Keywords: Anonymous authentication, Cloud services, Cryptography, Encryption, Group signatures, Privacy, Security.

1. Introduction

Cloud services are becoming indisputable parts of modern information and communication systems and step into our daily lives. Some cloud services such as Amazon's Simple Storage Service, Box.net, CloudSafe etc. use user identity, personal data and/or the location of clients.Therefore, these cloud computing services open a number of security and privacy concerns. The current research challenge in cloud services is the secure and privacy-preserving authentication of users. Users, who store their sensitive information like financial information, health records, etc., have a fundamental right of privacy. There are few cryptographic tools and schemes like anonymous authentication schemes, group signatures, zero knowledge protocols that can both hide user identity and provide authentication. The providers of cloud services need to control the authentication process to permit the access of only valid clients to their services. Further, they must be able to revoke malicious clients and reveal their identities.

In practice, hundreds of users can access cloud services at the same time. Hence, the verification process of user access must be as efficient as possible and the computational cryptographic overhead must be minimal.

We propose a novel security solution for cloud services that offers anonymous authentication based on group signatures. We aim mainly on the efficiency of the authentication process and user privacy. Our solution also provides the confidentiality and integrity of transmitted data between users and cloud service providers. Moreover, we implement our solution as a proof-of-concept application and compare the performance of our solution with related schemes. Our results show that our solution is more efficient than the related solutions.

The paper is organized as follow: The next section presents the related work. Then, we analyse cryptographic privacy-preserving schemes used in cloud computing. In section 4, we describe group signatures. In section5, wepresent our solution and we introduce our novel privacy-preserving cryptographic scheme for cloud services in section 6. Section 7contains our experimental results and the performance analysis and comparison. Finally, the conclusion of our work is presented.

2. Related work

Privacy-preserving cloud computing solutions have been developed from theoretical recommendations to concrete cryptographic proposals.

There are many works which deal with general security issues in cloud computing but only few works deal also with user privacy.

The authors [1] explore the cost of common cryptographic primitives (AES, MD5, SHA-1,RSA, DSA, and ECDSA) and their viability for cloudsecurity purposes. The authors deal with the encryption of cloud storage but do not mention privacy-preserving access to a cloud storage.

The work [2] employs a pairing based signature scheme BLS to make the privacy-preserving security audit of cloud storage data by the Third Party Auditor (TPA). The solution uses batch verification to reduce communication overhead from cloud server and computation cost on TPA side.Further, the paper [3] introduces the verification protocols that can accommodate dynamic data files. The paper explores the problem of providing simultaneous public auditability and data dynamics for remote data integrity check in Cloud Computing in a privacy-preserving way.These solutions [2] and [3] provide privacy-preserving public audit but do not offer the anonymous access of users to cloud services.

The work [4] establishes requirements for a secure and anonymous communication system that uses a cloud architecture (Tor and Freenet). Nevertheless, the author does not outline any cryptographic solution. Another non-cryptographic solution ensuring user privacy in cloud scenarios is presented in[5]. The authors propose a client-based privacy manager which reduces the risk of the leakage of user private information.In the paper [6], authors use a non-cryptographic approach to obtain the benefits of the public cloud storagewithout exposing the content of files. The approach is based on redundancy techniques including an information dispersal algorithm (IDA).Nevertheless, these solutions do not protect against the linkability of user sessions which can cause unauthorized user profiling.

Jensen et al. [7] propose an anonymous and accountable access method to cloud based on ring and group signatures. Nevertheless, their proposal uses a group signature scheme [8] which is inefficient because the signature size grows with the number of users.

The work [9] presents a security approach which uses zero-knowledge proofs providing user anonymous authentication. The main drawback of the proposal is a large communication overhead between a user and a cloud server due to the Fiat-Shamir identification scheme [10]. In the work [11], the author uses the CLsignature scheme [12] and zero-knowledge proofs of knowledge to achieve user's anonymous access to services like digital newspapers, digital libraries, music collections, etc.

The work [13] presents a cryptographic scheme to ensure anonymous user access to information and the confidentiality of sensitive documents in cloud storages. The work[14]deals with anonymity and unlinkability in cloud services by provided group signature schemes[15]. In the next section, we analyze the solutions [11], [13] and [14].

3. Performance analysis of cryptographic privacy-preserving solutions used in cloud computing

In this section, we investigate the current cryptographic solutions which provide the anonymous or pseudonymous access to cloud services and shared storages.We aim on the authentication phases used in privacy-preserving cloud services. In the following performance analysis, we take into account only expensive operations like bilinear pairings (p), modular exponentiation (e) and multiplication (m). According to the results of works [16], [17], we omit the fast operations like addition, subtraction or hash functions which have a minimal impact on the overall performance. The times of expensive pairing operations have been measured for example in [25].

Table 1shows the performance analysis of the Blantom solution [11], the Lu et al. solution[13], the Chow et al. solution [14]and our scheme described in Section6. Blantom in [11]proposes a solution

using the CL signatures [12]. To establish anonymous authentication, the CL signature is combined with a Zero Knowledge Proof ofKnowledge (ZKPK) protocols. The computational complexity of Blantom solution depends on the subscription type and is variable. Lu et al. [13]propose a pairing-based cryptographic scheme ensuring anonymous authentication of users accessing cloud services. A user has to sign a challenge received from a server and then he/she sends it back to verify it. Chow et al. [14]employ group signature schemes proposed by Boyen and Waters in [15]and [18](BW schemes). The BW scheme [18]is used to make a group signature which provides the anonymous authentication of users. Nevertheless, these solutions have 6 pairing operations in verification.In the next section, we present our solution that does not use expensive pairing operations.

Solutions:	Communication overhead	Signing (Authenticate)	Verification
Blantom solution [11]	various	various (approx. 30p +31e + 12m)	6p + 17e + 5m
Lu et al. solution [13]	5 elements	14e + 10m	6p + 1e + 2m
Chow et al. solution [13]	6 elements	14e + 15m	6p + 1e + 6m
Our solution	12 elements	10e + 8m	12e + 6m

Table 1. Performance Analysis of Solutions in Cloud Computing.

4. Group signatures as a basic part in privacy enhancing cloud services

Group signature schemes are used in many privacy enhancing cryptographic protections that are applied in cloud services. Group signatures were introduced by Chaum and Heyst [8] in 1991. Their main purpose is to allow members of a group sign messages on behalf of the group. Every group member can sign a message by own group member secret key *gsk[i]* that is usually issued by a group manager. A verifier checks the validity of the signature with a group public key *gpk*. The verifier is able to verify that the signer is indeed the member of the group while the signer's identity is not released. The identities of the members are traceable only in certain circumstances, e.g. breaking the rules. Revocation can be done by the group manager or a revocation manager who owns group manager's secret key *gmsk*. The group signature schemes usually employ the following entities:

• **Group manager** – this entity adds group members into a group, and generates and issues the secret keys of group members.

• **Revocation manager** – this entity disclosures the identity of dishonest members.

• **User** – a group member who owns the group member secret key *gsk[i].* The user can sign a message on behalf of the group.

• **Verifier** – this entity verifies the validity of the signature by using the group public key *gpk*.

Currently, there are many variants of group signatures schemes which differ mainly in their properties such as the level of anonymity, security, efficiency and the length of signature. Group signatures can be understood as a subset of attribute authentication systems, which contain only one attribute representing a membership in a group. Group signatures schemes usually provide the following properties:

• **Unforgeability** - only an unrevoked group member can create a valid signature on behalf of the group.

• **Anonymity**– a verifier is not able to determine the identity of a signer.

• **Complete anonymity** – if an attacker obtains a valid signature and knows *gpk* and all keys of group members' *gsk[i],* he is not able to determine the identity of a signer.

• **Traceability** - all members can be tracked by the group manager or the revocation manager by member's signed message.

• **Unlinkability**- a verifier and other members are not able to link two signatures which have been signed by one member of the group.

• **Coalition-resistance**- it is impossible to create a valid signature by a subgroup of users.

• **Exculpability** – even group manager is not able to create a valid signature of a group member.

• **Correctness** - every correct signature of the group member has to be always accepted during verification.

• **Revocation**–a revoked member is not able to create valid signatures on behalf of the group.

• **Differentiation of group members** - all members of the group must have a different *gsk[i].*

• **Immediate-revocation** – if a group member is revoked, his capability of creating the group signatures is disabled immediately.

5. Our solution

In this chapter, we introduce our security solution for privacy-preserving cloud services. We outline our system model depicted in Figure 1, security requirements, cryptography background and cryptographic protocols.

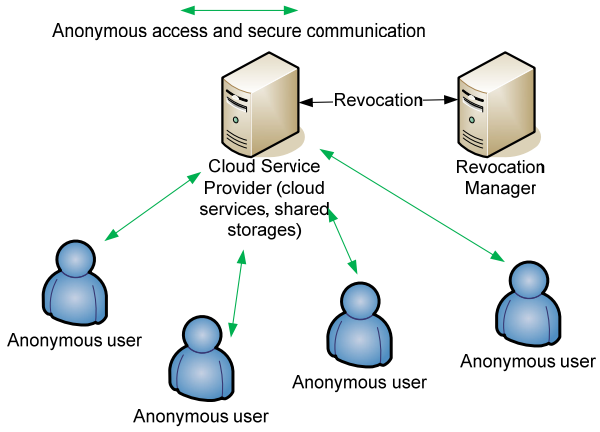

Figure 1. System model.

5.1 System Model

Our solution consists of three fundamental parties:

Cloud Service Provider (CSP). CSP manages cloud services and shared storages. CSP isusually a company which behaves as a partly trusted party. CSP provides cloud services, authenticates users when they access a cloud service. CSP also issues access attributes to users. Nevertheless, when CSP needs to revoke and identify a malicious user then CSP must collaborate with a revocation manager.

Revocation Manager (RM). RM is a partly trusted party, e.g. government authority, who decides if the revocation of a user identity is rightful or not. Only the cooperation between CSP and RM can reveal the user identity. RM also cooperates with CSP during user registration when the user's access attributes are issued.

User (U). U is an ordinary customer who accesses into a cloud and uses cloud services, shared storages, etc. Users are anonymous if they properly follow the rules of CSP. To increase security, users use tamper-resistant devices or protected local storages.

5.2 Requirements

Our solution provides the following security requirements:

• *Anonymity*. Every honest user stays anonymous when uses cloud services. User identities are hidden if users behave honestly and do not break rules.

• *Confidentiality*. Every user's session to CSP is confidential. No one without a secret session key is able to obtain data transmitted between U and CSP.

• *Integrity*. Data sent in user's session cannot be modified without a secret session key

• *Unlinkability.* The user's sessions to cloud services are unlinkable. No one besides CSP collaborating with RM is able to link two or more sessions between a certain U and CSP.

• *Untraceability.* Other users are unable to trace user's authentication and concrete users' communication.

• *Revocation.* Every user can be revoked by the collaboration of CSP and RM.

5.3 Cryptography Used

In our solution, we use discrete logarithm commitments described in the work [19]. We have

transformed the scheme [19] into a group signature scheme mode.Further,the solution employs Σ-protocols [20]to prove of discrete logarithm knowledge, representation and equivalence [21]. To revoke a user, we use the Okamoto-Uchiyama Trapdoor One-Way Function described in [22]. For more details about the used basic cryptographic blocks see prior works[19]and[23].

6. Ourproposed protocol

Our protocol consists of five phases: initialization, registration, anonymous access, secure communication and revocation. The basic principle of the proposed protocol is depicted in Figure 2.

• {tc "1 The Basic Principle of the Proposed Protocol." \f f}

6.1 Initialization

The initialization phase is run by Cloud Service Provider (CSP) and Revocation Manager (RM). CSP generates a group *H* defined by a large prime modulus p, generatorsh_1, h_2 of prime order q and $q|p-1$. CSP generates a RSA key pair and stores own private keyK_{CSP}.

M generates a group *G* defined by a large modulus $n = r^2 s$ where $r = 2r' + 1$, $s = 2s' + 1$ and $r, s, r', s'$are large primes. RM also generates a generator $g_1 \in_R \mathbb{Z}_n^*$of order$ord(g_1 mod r^2) = r(r-1)$ in $\mathbb{Z}_{r^2}^*$and $ord(g_1) = rr's'$ in $\mathbb{Z}_n^*$and randomly chooses secret values S_1, S_2, S_3. RM computes authentication proof $A_{proof} = g_1^{S_1} mod\ n$which is public and common for all entities in system. In our solution, the RM is able to issue more types of authentication proofs$A_{proof}^1 \dots A_{proof}^N$derived from $S_1^1 \dots S_1^N$that are related to different user rights in cloud services.

Finally, RM computes generators $g_2 = g_1^{S_2} mod\ n$and $g_3 = g_1^{S_3} mod\ n$and stores secret valuesr, sas revocation keyK_{RK}.

All public cryptographic parameters $q, p, n, g_1, g_2, g_3, h_1, h_2, A_{proof}$ are published and shared.

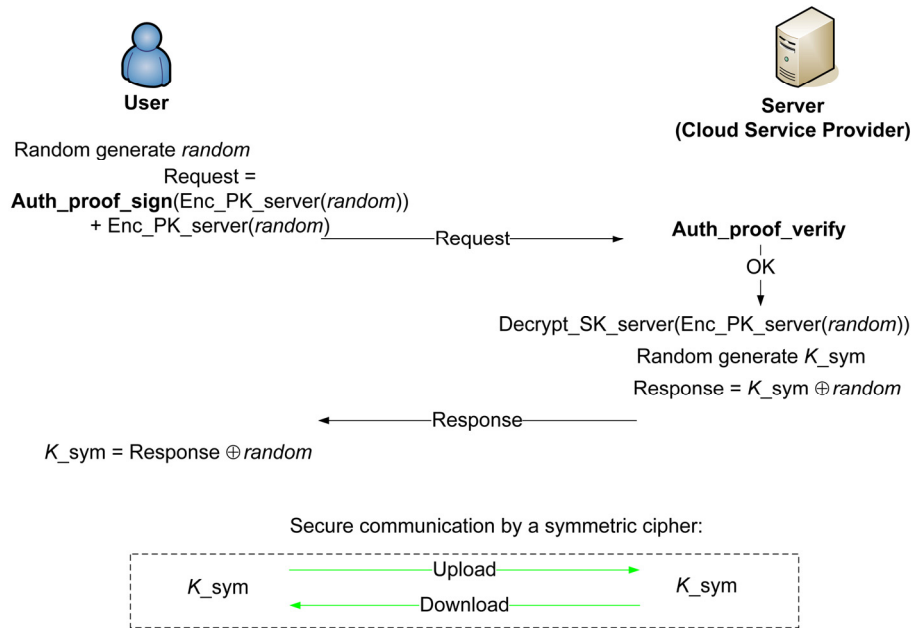

Figure 2. The basic principle of the proposed protocol.

6.2 Registration

In the registration phase, a user registers and requests a user master key which they use in anonymous access to cloud services.

Firstly, U must physically register on CSP. CSP checks user's ID. Then, U generates secret values $\omega_1, \omega_2$and makes the commitment: $C_{CSP} = h_1^{\omega_1} h_2^{\omega_2} \; mod\, p$. U digitally signs C_{CSP}, e.g. by RSA, and sends this signature $Sig_U(C_{CSP})$with the construction of correctness proof PK$\{\omega_1, \omega_2 : C_{CSP} = h1\omega1h2\omega2$ to CSP, by notation of Camenisch and Stadler [21]. CSP checks the user's proof and the signature. Then, CSP stores the pair$C_{CSP}, Sig_U(C_{CSP})$, signs the commitment $Sig_{CSP}(C_{CSP})$and sends it back to U.

Secondly, U requests a user master key from RM. U computes$A'_{proof} = g_1^{\omega_1} g_2^{\omega_2} \; mod\, n$and sends it with $C_{CSP}, Sig_{CSP}(C_{CSP})$ and the construction of correctness proof PK$\{\omega_1, \omega_2 : C_{CSP} = h_1^{\omega_1} h_2^{\omega_2} \wedge A'proof=g1\omega1g2\omega2$ to RM. RM checks the proof, CSP's signature$Sig_{CSP}(C_{CSP})$ and computes a secret contribution ω_{RM} such that $A_{proof} = g_1^{\omega_1} g_2^{\omega_2} g_3^{\omega_{RM}} \; mod\, n$holds. After this step, U obtains own user master key K_U which is triplet$(\omega_1, \omega_2, \omega_{RM})$. U gets value ω_{RM} only with cooperation with RM which knows the factorization of n. To prevent the collusion attack, user'sω_1, ω_2 is not visible outwardly to a user because ω_1, ω_2 is stored in a tamper-resistant memory. This device which stores the user secret key should be also protected against a key estimation by side channel attacks, such as in [24]. Further, U cannot make own user master key because only RM knowsK_{RK}. Any honest user can repeat the request for the user master key or demand other authentication proofs if CSP agrees with that.

6.3 Anonymous Access

In this phase, the i-th userU_i anonymously accesses Cloud Service Provider (CSP). This phase consists of two-messages used to authenticate U_i and establish a secret key betweenU_i and CSP.

• U_i generates a random value$random \in_R \{0,1\}^{l_{sym}}$. The parameter l_{sym} denotes the size of a shared secret key for the symmetric cipher.

• U_i encrypts$random$ by the RSA public key of CSP.

•The encrypted Enc_PK_server($random$) is signed by the **Auth_proof_sign** algorithm in the group signature modewhich ensures user anonymous authentication. We assume that cryptographic parameters such as $q, p, n, g_1, g_2, g_3, h_1, h_2$and authentication proof $A_{proof} = g_1^{\omega_1} g_2^{\omega_2} g_3^{\omega_{RM}} \; mod\, n$ are made public and $\mathcal{H}$ is a secure hash function. To prove the knowledge of the secret user key and sign$random$, U_i performs the **Auth_proof_sign** algorithm:

$$K_S \in_R \{0,1\}^l$$
$$A = A_{proof}^{K_S} \; mod\, n$$
$$C_1 = g_3^{K_S \omega_{RM}} \; mod\, n$$
$$C_2 = g_3^{K_S} \; mod\, n$$
$$r_1, r_2 \in_R \{0,1\}^{m+k+3l}$$
$$r_3 \in_R \{0,1\}^{m+k+4.5l}$$
$$r_s \in_R \{0,1\}^{m+k+l}$$
$$\overline{A_{proof}} = g_1^{r_1} g_2^{r_2} g_3^{r_3} \; mod\, n$$
$$\bar{A} = A_{proof}^{r_S} \; mod\, n$$
$$\overline{C_1} = g_3^{r_3} \; mod\, n$$
$$\overline{C_2} = g_3^{r_S} \; mod\, n$$
$$c = \mathcal{H}(\text{Enc\_PK\_server}(random), A, \bar{A}, \overline{A_{proof}}, C_1, C_2, \overline{C_1}, \overline{C_2})$$
$$z_1 = r_1 - cK_S\omega_1$$
$$z_2 = r_2 - cK_S\omega_2$$
$$z_3 = r_3 - cK_S\omega_{RM}$$
$$z_S = r_S - cK_S$$

Finally, the signature elements $A, \bar{A}, \overline{A_{proof}} C_1, C_2, \overline{C_1}, \overline{C_2}, z_1, z_2, z_3, z_S$, Enc_PK_server($random$) are sent to CSP as a request message.

• CSP verifies the signed request message that consists of the signature elements: Enc_PK_server($random$),$A, \bar{A}, \overline{A_{proof}}$,

$C_1, C_2, \overline{C_1}, \overline{C_2}, z_1, z_2, z_3, z_S$. Then, CSP does the **Auth_proof_verify** algorithm:

$$C_1 \not\equiv C_2^{rev} \bmod n$$
$$\overline{A_{proof}} \equiv A^e g_1^{z_1} g_2^{z_2} g_3^{z_3} \bmod n$$
$$\bar{A} \equiv A^e A_{proof}^{z_S} \bmod n$$
$$\overline{C_1} \equiv C_1^e g_3^{z_3} \bmod n$$
$$\overline{C_2} \equiv C_2^e g_3^{z_S} \bmod n$$

If above equations hold then CSP continues in the next step. Otherwise, CSP stops the algorithm.

• CSP decrypts a value Enc_PK_server($random$) by its RSA private key to obtain$random$.

• CSP randomly generates shared secret key *K*_sym and uses eXclusive OR (XOR) function to compute $random \oplus$*K*_sym.

• CSP sends a response message ($random \oplus$*K*_sym) back toU_i.

6.4 Secure Communication

If the anonymous access phase is successful, the user $U_i$can upload and download data from CSP. Data confidentiality and integrity are secured by a symmetric cipher. We propose to use AES which is well know cipher and is supported by many types of software and hardware platforms. To encrypt and decrypt transmitted data, $U_i$and CSP use the AES secret key *K*_sym established in the previous phase.

6.5 Revocation

Depending on the case of rule breaking, the revocation phase can revoke a user and/or user anonymity.

If users misuse a cloud service, they get revoked by RM. Because RM knows the factorization of$n$, RM is able to extractω_{RM}. Firstly, RM extracts the random session value K_S from C_2 and the secret RM contribution value ω_{RM} fromC_1.

Then, RM publishes ω_{RM} into a public blacklist. If the user uses revoked key then the equation $C_1 \equiv C_2^{\omega_{RM}} \bmod n$holds and the user access to cloud services is denied.

If a malicious user breaks the rules of CSP, this user can be identified by the collaboration of RM and CSP. Firstly, RM extracts ω_{RM} from the suspected session received by CSP. Then, RM finds the corresponding C_{CSP} in the database. If CSP provides to RM the explicit evidence of user's breach, then RM sends $C_{CSP}$to CSP. CSP is able to open the identity of a user from database but only with RM's help.

7. Experimental results

In this section, we outline the experimental results of our solution. We compare our solution with related solutions and output the performance evaluation.

7.1 Performance Evaluation of Our Solution

We have implemented our proposed solution in JAVA. In practice, we expect that U as an end node uses devices with reasonable computational power such as a personal computer, a laptop, a tablet or a smartphone. On the other hand, we assume that CSP keeps servers with sufficient computational capacity to ensure hundreds sessions with end nodes in real time. We have tested our solution on a machine with Intel(R) Xeon(R) CPU X3440 @ 2.53GHz, 4 GB Ram. In our a proof-of-concept implementation, we choose the 1024-bit length of modulo. The main important part of our solution is the Anonymous Access phase. In this phase, a user (U) communicate with a Cloud Service Provider (CSP). The computation process on the user side is marked as the Sing/Authenticate process. The computation process on the CSP side is marked as the Verify process. We have measured the total time of the Sing/Authenticate process and the Verify process. In the Verify process, Table 2 shows two scenarios: with an empty black list and with the black list that contains the revoked values *rev* = 10. The influence of the size of blacklist on the total time of the Verify process is depicted in Figure 3.

Sessions [#]	Sign/Authenticate Total time [ms]	Verify	Verify with *rev* = 10
1	54	70	106
10	526	721	920
20	1042	1272	1891
50	2504	3328	4091

Table 2. Performance Evaluation of Our Solution.

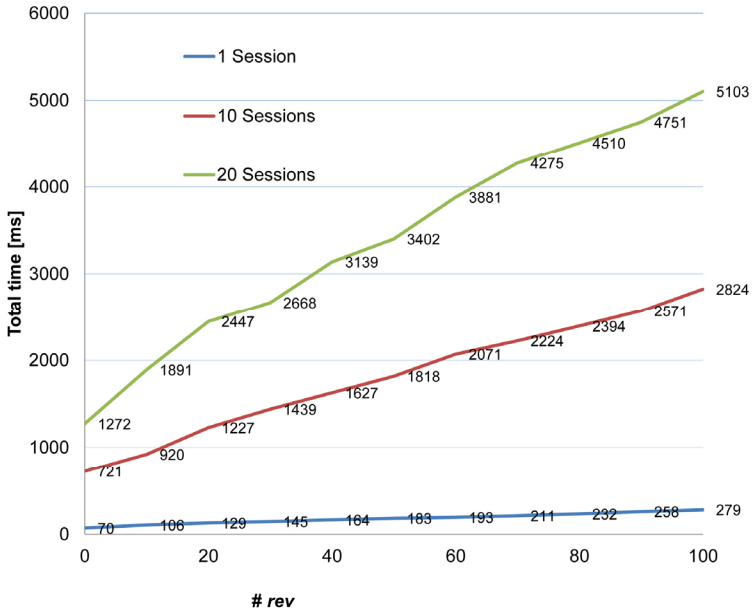

Figure 3. Influence of the length of the blacklist on the total time of verification.

7.2 Comparison with Related Work

We compare our Anonymous Access phase with the authentication phase of related solutions: Blantom solution [12], Lu et al. solution [13]and Chow et al. solution [14]. To ensure objectivity, we compare the number of atomical cryptographic and math operations for each solution.

Firstly, we compare the Sign/Authenticate process that runs on the user side. In the Sign/Authenticate process, Lu et al. solution [13]takes 14 exp + 10 mul, Chow et al. solution [14]takes 14 exp + 15 mul and Blantom's solution [12]takes tens of pairing and exponentiation operations. The number of operations in Blantom's solution [12] depends on the subscription type and is variable. Our Sign/Authenticate process takes only 8 exp + 5 mul and is the most efficient from compared solutions.

The Verify process on the CSP side has 10 exp + 6 mul in our solution. We emphasize that our solution has 0 paring operations. Lu et al. solution [13], Chow et al. solution [14]and Blantom solution [12]are pairing based and contain 6 pairing operations in the Verify process.Figure 4 depicts the performance of the verifyprocess of our and related solutions. The verify process of our solution is more efficient than related solutions in this comparison and takes only 28 % of the total time of Lu et al. solution [13]or Chow et al. solution [14].

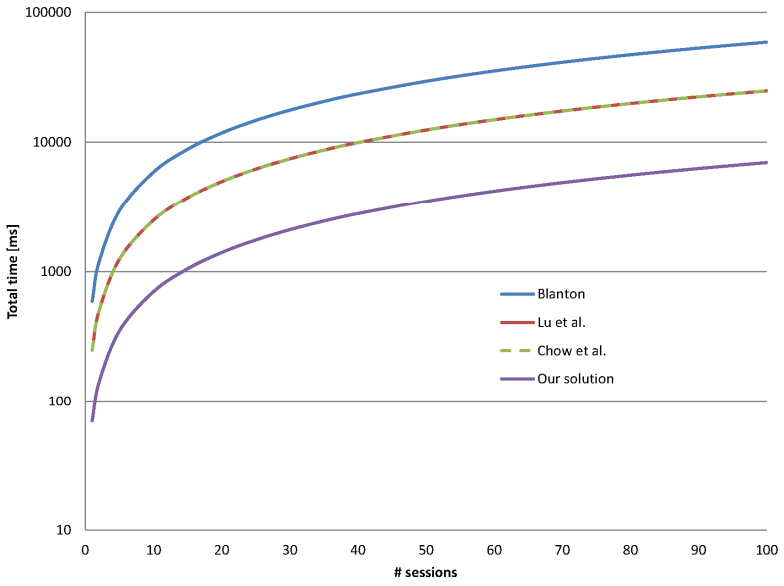

Figure 4. Performance of the verify process.

7.3 Comparison of our group signature scheme with the related work

In this section, we analyze group signature schemes from open literatureandcompare them with ourgroup signature schemeused in our proposal, see Table 3.In the following part, we analyze group signature schemes and describe their evolution.

Group signatures were introduced and first four schemes were presented in the work CHH91[8]in 1991. The main disadvantage of these schemes is long sizes of a group public key *gpk* and a signature. Sizes depend on the number of members in a group. If a new member is added to the group, it is necessary to modify *gpk*. These deficiencies are very impractical for large groups of members. Therefore, these schemes are not suitable in cloud services. In the work CS97[26], published in 1997, authors propose a scheme which using the constant size of *gpk* and signatures. New members can be added to the group without the need to generate a new key pair *gpk* and *gsk[i]*. The paper ACJT00[27], introduced in 2000, presents an efficient scheme which is resistant of coalition, i.e. it is impossible for a subset of the group members including the group manager to create a valid signature. The disadvantage of the scheme is missing of the revocation of group members and prevention to a revoked member generates the valid signatures on behalf of the group. The work AST02 [28], published in 2002, is based on the scheme ACJT00 [27] and adds the revocation of the group members without using a time stamp. This approach keeps a constant length of a signature, i.e. this length does not increase linearly with the number of revoked members. However, the scheme has more operations in signing and verification phases than related schemes.The scheme TX03 [29], published in 2003, provides the dynamic revocation of group members. Revoked members are no longer able to create a valid signature. On the other hand, the disadvantage is that, *gpk* has to be recalculated when a member is added to the group or remove from the group. This approach is highly inefficient in the real time systems working with large groups. The schemes BS04 [30] and BBS04 [31], published in 2004, allow to create short group signatures. These schemes are based on bilinear maps and produce short signatures which are suitable in systems where bandwidth is restricted.Unless as the previous schemes that are secure in the random oracle model, the scheme BMW03 [32], introduced in 2003, is secure in the standard model. Nevertheless, the scheme is designed for the static

and small groups of users. Therefore, this scheme is not proper for cloud services.

The scheme ACHM05 [33], introduced in 2005, is provable secure in the standard model and works with dynamic groups. The scheme provides anonymity, unforgeability, untraceability and exculpability, and is secure against a non-adaptive adversary who does not have *gsk[i]* of group members. The scheme BW06 [15] provides the provable security in the standard model. But, the size of the signature depends on the size of the group. The newer scheme BW07 [18], introduced in 2007, produces shorter and almost constantly sized signature in comparison with the previous schemes. The length of a signature increases logarithmically as the size of the group.

The scheme LCSL07 [34] produces short signatures with constant lengths. This scheme offers full anonymity and full traceability, and the public key and signatures are shorter than in the previous schemes. The scheme G07 [35], published in 2007, ensures full anonymity in the standard model. The scheme is based on bilinear groups and produces the constant lengths of keys and signatures. The scheme also supports the dynamic addition of new members to the group.

We compare our scheme with the group signature schemes in Table 3.Our scheme is based on non-bilinear assumptions and has only 10 exponentiations and 8 multiplications in the verification phase. Our scheme clearly outperforms the related schemes. The operations are abbreviated as bp– bilinearpairings, e - exponentiation, mul– multiplication,div - division, add – addition (subtraction), H–hash, k- length of identities in bits,m - length of message in bits, RL–members in a revocation list, EF - efficiently computable isomorphism from G2 toG1, T – the total time of a period.

Scheme	Signing operations	Verification operations	Size of signature	Size of group public key	Efficiency	Security model	Type
ACJT00 [27]	14e + 1H+ 9mul + 2div + 6add	15e + 1H + 9mul + 2div + 4sum	8696 b	6144 b	Constant *gpk* and sign.	Random Oracle	Non-bilinear
NS04 [36]	3e + 32mul + 14add + 1H	3p + 2e + 14mul + 8add + 1H	4776 b	2904 b	Constant *gpk* and sign.	Random Oracle	Bilinear
BBS04 [31]	3p + 12e + 10mul + 8add + 1H	5p+12e +7mul + 1div + 2add + 1H	3|G_1| + 6|Zp| 1 553 b	6|G_1| 1026 b	Constant *gpk* and sign.	Random Oracle	Bilinear
BS04 [30]	3p + 2EF + 8e +8mul + 3add + 2H	(6 +|*RL*|)p + 8e + 4mul + 2div +2H + 2EF	2|G_1|+5|Zp| 1 192 b	3|G_1| 513 b	Constant *gpk* and sign.	Random Oracle	Bilinear
ACHM05[33]	8e + 2div + 2add	10p + 1e + 3mul	6|G_1|+2|G_2| 2052 b	2|G_1| + 4|G_2| + |G_T|	Constant *gpk* and sign.	Standard	Bilinear
BW06[15]	(5k + m + 5)e + (4k + m + 4)mul + (2k-1)add	(3+2*k*)p + *m*e + (*m*+*k*)mul	(2k + 3)|G|	(*k* + *m* + 3)|G|+|G_q|+|G_T|	Logarithmic *gpk* and sign.	Standard	Bilinear
ZL06 [37]	2p + 17e + 17mul + 7sum + 2div + 1H	(3 + *RL*)p +17e +9mul + 2div + 1H	8|Zp|+5|G| 2215 b	(3 + T)|G|	Constant sign.	Random oracle	Bilinear
BW07 [18]	(12 + 2*m*)e + (11 + 2*m*)mul	6p +(3 + *m*)e + (4 + *m*)mul	6|G| 1026 b	(4 + *m*)|G| +|G_q| + |G_T|	Logarithmic *gpk* a constant sign.	Standard	Bilinear
LCSL07 [34]	12e + 10mul + 1div + 1H + 1add	6p + 3e + 4mul	5|G|	3|G| + |Gq|	Constant *gpk* and sign.	Standard	Bilinear
Our scheme	10e + 8m	12e + 6m	8835 b	5950 b	Constant	Random Oracle	Non-bilinear

Table 3. Comparison of Group Signatures Schemes with Our Solution.

8. Conclusion

In this paper, we present our novel security solution for privacy-preserving cloud services. We propose the non-bilinear group signaturesscheme to ensure the anonymous authentication of cloud service clients. Our novel solution offers user anonymity in the authentication phase, data integrity and confidentiality and the fair revocation process for all users. Users use tamper resistant devices during the generation and storing of user keys to protect against collusion attacks.

Our authentication phase, which is based on the non-bilinear group signature scheme, is more efficient than related solutions on the client side and also on the server side due to missing expensive bilinear pairing operations and fewer exponentiation operations.Thus, cloud service providers using our solution can authenticate more clients in the same time. We also analyze related group signature schemes. The group signature scheme used in our solution is more efficient than related group signature schemes in the verification phase and provides the efficient privacy-preserving access to cloud services.

Acknowledgments

This research work is funded by project SIX CZ.1.05/2.1.00/03.0072; the Technology Agency of the Czech Republic projects TA02011260 and TA03010818; the Ministry of Industry and Trade of the Czech Republic project FR-TI4/647.

References

[1] Y. Chen and R. Sion, “On securing untrusted clouds with cryptography,”in Proceedings of the 9th annual ACM workshop on Privacy in the electronic society, New Your k,ACM, 2010, pp.109–114.

[2] C. Wang, et al., “Privacy-preserving public auditing for data storage security in cloud computing,” in INFOCOM,2010 Proceedings IEEE, San Diego march 2010, pp. 1–9.

[3] Q. Wang et al. “Enabling public auditability and data dynamics for storage security in cloud computing,” in Parallel and Distributed Systems, IEEE Transactions on, vol. 22, no. 5,IEEE, 2011. pp. 847–859.

[4] R. Laurikainen, “Secure and anonymous communication in the cloud, ”in Aalto University School of Science and Technology, Department of Computer Science and Engineering, Tech. Rep. TKK-CSE-B10, 2010, pp. 1-5

[5] M. Mowbray and S. Pearson, “A client-based privacy manager for cloud computing,” in Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middle waRE, ser. COMSWARE ’09, New York, ACM, 2009, pp. 5:1–5:8.

[6] E.M. Hernandez-Ramirez et al. “A Comparison of Redundancy Techniques for Private and Hybrid Cloud Storage,” in JART Journal of Applied Research and Technology, vol. 10, no. 6, pp. 1-9, 2012.

[7] M. Jensen et al., “Towards an anonymous access control and accountability scheme for cloud computing,” in Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, Miami, IEEE. 2010, pp. 540 –541.

[8] D. Chaum and E. Van Heyst, “Group signatures,” in Advances inCryptologyEUROCRYPT91. 1991, pp. 257–265.

[9] P. Angin et al., “An entity-centric approach for privacy and identity management in cloud computing,” in Reliable Distributed Systems, 201029th IEEE Symposium on, New Delhi, IEEE. 2010, pp. 177–183.

[10] A. Fiat and A. Shamir, “How to prove yourself: practical solutions to identification and signature problems,” in Advances in Cryptology-Crypto86. 1987, pp. 186–194.

[11] M. Blanton, “Online subscriptions with anonymous access,” in Proceedings of the 2008 ACM symposium on Information, computer and communications security, ser. ASIACCS ’08, New York, ACM. 2008, pp. 217–227.

[12] J. Camenisch and A. Lysyanskaya, “Signature schemes and anonymous credentials from bilinear maps.” in Advances in Cryptology– CRYPTO2004, 24th Annual International Cryptology Conference, Santa Barbara, California, USA. 2004, pp. 56–72.

[13] R. Lu et al., “Secure provenance: the essential of bread and butter of data forensics in cloud computing,” in Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS ’10, New York, ACM, 2010, pp. 282–292.

[14] S. Chow et al., “Spice–simple privacy-preserving identity-management for cloud environment,” in Applied Cryptography and Network Security. 2012, pp. 526–543.

[15] X. Boyen and B. Waters, “Compact group signatures without random oracles,” in Advances in Cryptology-EUROCRYPT 2006. 2006, pp. 427–444.

[16] L. Malina and J. Hajny, “Accelerated modular arithmetic for low-performance devices,” in Telecommunications and Signal Processing(TSP), 2011 34th International Conference on, Budapest, IEEE. 2011, pp. 131–135.

[17] L. Malina and J. Hajny, “Efficient modular multiplication for programmable smart-cards.” in TelSys. Telecommunication Systems, pp.1-8. 2013.

[18] X. Boyen and B. Waters, “Full-domain subgroup hiding and constant-size group signatures,” Public Key Cryptography–PKC 2007.Beijing, China. 2007, pp. 1–15.

[19] J. Hajny and L. Malina, “Unlinkable attribute-based credentials with practical revocation on smart-cards,” in Proceedings of the 11th international conference on Smart Card Research and Advanced Applications, ser. CARDIS’12. Springer-Verlag, 2013, pp. 62–76.

[20] R. Cramer, “Modular design of secure, yet practical cryptographic protocols,” Ph.D. dissertation, University of Amsterdam, 1996.

[21] J. Camenisch and M. Stadler, “Proof systems for general statements about discrete logarithms,” Tech. Rep., 1997.

[22] T. Okamoto and S. Uchiyama, “A new public-key crypto system as secure as factoring,” in Advances in Cryptology - EUROCRYPT 98, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, vol.1403, pp. 308–318, 1998.

[23] J. Hajny and L. Malina, “Practical revocable anonymous credentials,” in Communications and Multimedia Security, Canterbury, UK. 2012, pp. 211–213.

[24] Z. Martinasek et al., “Optimization of differential poweranalysis,” Przeglad elektrotechniczny,vol. 87, no. 12, pp. 140–144, 2011.

[25] L. Martínez-Ramos et al., "Achieving Identity-Based Cryptography in a Personal Digital Assistant Device." JART. Journal of Applied Research and Technology, vol. 9. no. 3, pp. 1-11, 2011.

[26] J. Camenisch and M. Stadler, “Eficient group signatures chemes for large groups,” in Advances in Cryptology — CRYPTO ’97, California, USA. 1997, pp. 410-424, 2011.

[27] G. Ateniese et al., “A practical and provably secure group signature scheme,” in proceedings of CRYPTO ’00. 2000, pp. 255–270.

[28] G. Ateniese et al., “Quasi-efficient revocation in group signatures” in proceedings of Financial Cryptography ’02. 2002, pp. 183–197.

[29] G. Tsudik and S. Xu, “Accumulating composites and improved group signing,” in proceedings of ASIACRYPT ’03. 2003, pp. 269–286.

[30] D. Boneh and H. Shacham. “Group signatures with verifier-local revocation,” in Conference on Computer and Communications Security, Proceedings of the 11th ACM conference on Computer and communications security, Washington DC, USA. 2004, pp. 168 – 177.

[31] D. Boneh et al.,“Short group signatures,” in Advances in Cryptology – CRYPTO 2004, Santa Barbara, California, USA. 2004, pp. 41-55.

[32] M. Bellare et al., “Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions,” in Advances in Cryptology - EUROCRYPT '03, Warsaw, Poland. 2003, pp. 614-629.

[33] G. Ateniese et al., “Practical group signatures without random oracles,” IACR Cryptology ePrint Archive.pp. 1-31, 2005.

[34] X. Liang et al., “Short group signature without random oracles,” in Information and Communications Security, Zhengzhou, ICICS, China. 2007, pp. 69-82.

[35] J. Groth,“Fully anonymous group signatures without random oracles,” in Advances in Cryptology – ASIACRYPT 2007, Kuching, Malaysia. 2007, pp. 164-180.

[36] L. Nguyen and R. Safavi-Naini, “Efficient and Provably Secure Trapdoor-free Group Signature Schemes from Bilinear Pairings,”in Advances in Cryptology - ASIACRYPT 2004, Jeju Island, Korea. 2004, pp. 372-386.

[37] S. Zhou and D. Lin, “A shorter group signature with verifier-location revocation and backward unlinkability,”CryptologyePrint Archive, Report 2006/100. 2006, pp. 1-10.