# Dynamic Load Balancing For Cloud Computing Using Virtual Machine

Srinivas Jangapalli[1], sateesh Nagavarapu[2]

[1]M Tech in CSE Malla reddy Institute of Technology, Hyderabad, Telangana, India.

[1] *j.seenu2@gmail.com*

[2]*Asst. Prof in CSE Department in Malla reddy Institute of Technology, Hyderabad, Telangana, India.*

[2]sateeshnagavarapu@gmail.com

**Abstract—In this paper, we present a novel privacy-preserving security solution for cloud services. We deal with user anonymous access to cloud services and shared storage servers. Our solution provides registered users with anonymous access to cloud services. Our solution offers anonymous authentication. This means that users' personal attributes (age, valid registration, successful payment) can be proven without revealing users' identity. Thus, users can use services without any threat of profiling their behavior. On the other hand, if users break provider's rules, their access rights are revoked. We analyze current privacy preserving solutions for cloud services and outline our solution based on advanced cryptographic components. Our solution offers anonymous access, unlinkability and the confidentiality of transmitted data. Moreover, we implement our solution and we output the experimental results and compare the performance with related solutions.**

**Keywords—Authentication, Cloud Computing, Cryptography, Encryption, Privacy, Security.**

## I. INTRODUCTION

EMERGING 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. 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 analyze cryptographic privacypreserving schemes used in cloud computing. In section IV.,

we introduce our novel privacy-preserving security solution for cloud services. Section V. contains our experimental results. Finally, the conclusion of our work is presented.

## II. 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 cloud security 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. Nevertheless, the solution does not protect against the linkability of user sessions which can cause unauthorized user profiling.

Jensen et. at. [6] propose an anonymous and accountable access method to cloud based on ring and group signatures.

Nevertheless, their proposal uses a group signature scheme [7] which is inefficient because the signature size grows with the number of users. The work [8] 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 [9]. In the work [10], the author uses the CL signature scheme [11] and zeroknowledge proofs of knowledge to achieve user's anonymous access to services like digital newspapers, digital libraries, music collections, etc. The work [12] presents a cryptographic scheme to ensure anonymous user access to information and the confidentiality of sensitive documents in cloud storages.

The work [13] deals with anonymity and unlinkability in cloud services by provided group signature schemes [14]. We analyze the solutions [10], [12] and [13] in the next section.

## III. 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 prior works [15], [16], we omit the fast operations like addition, subtraction or hash functions which have a minimal impact on the overall performance.

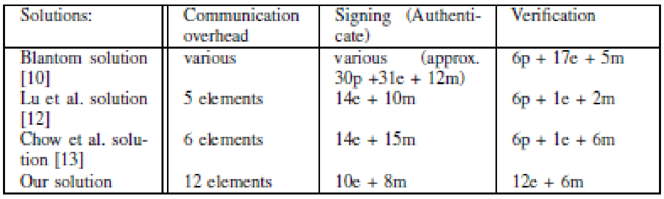
Table I shows the performance analysis of the Blantom solution [10], the Lu et al. solution [12], the Chow et al. solution [13] and our solution described in Section IV. Blantom in [10] proposes a solution using the CL signatures [11]. To establish anonymous authentication, the CL signature is combined with a Zero Knowledge Proof of Knowledge (ZKPK) protocols.

The computational complexity of Blantom solution depends on the subscription type and is variable. Lu et al. [12] 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. [13] employ group signature schemes proposed by Boyen and Waters in [14] and [17] (BW schemes). The BW scheme [17] 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.

## IV. OUR SOLUTION

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

TABLE I
PERFORMANCE ANALYSIS OF SOLUTIONS IN CLOUD COMPUTING.



| Solutions: | Communication overhead | Signing (Authenticate) | Verification |
|---|---|---|---|
| Blantom solution [10] | various | various (approx. 30p +31e + 12m) | 6p + 17e + 5m |
| Lu et al. solution [12] | 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 |

### A. System Model

Our solution consists of three fundamental parties:

- Cloud Service Provider (CSP). CSP manages cloud services and shared storages. CSP is usually 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.

### B. 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.

C. Cryptography Used

In our solution, we use discrete logarithm commitments described in prior work [18]. Further, the solution employs _protocols [19] to prove of discrete logarithm knowledge, representation and equivalence [20]. To revoke a user, we use the Okamoto-Uchiyama Trapdoor One-Way Function described in [21]. For more details about the used basic cryptographic blocks see prior works [22], [18].

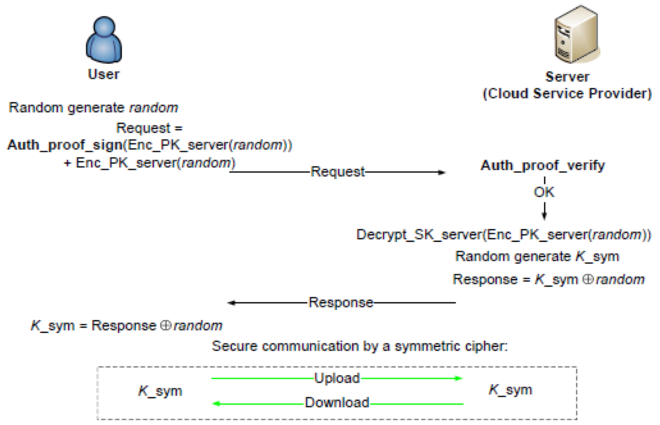


Fig. 1. The Basic Principle of the Proposed Protocol.

D. Proposed 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 Fig. 1.

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, generators h1; h2 of prime order q and qjp □ 1. CSP generates a RSA key pair and stores own private key KCSP .

RM generates a group G defined by a large modulus
n = r2s where r = 2r0 + 1; s = 2s0 + 1 and r; s; r0; s0 are
large primes. RM also generates a generator g1 2R Z_
n of order ord(g1 modr2) = r(r □ 1) in Z_
r2 and ord(g1) = rr0s0 in Z_ n and randomly chooses secret values S1; S2; S3. RM computes authentication proof Aproof = gS1 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 A1proof :::AN proof derived from S1
1 :::SN 1 that are related to different user rights in cloud services. Finally, RM computes generators g2 = gS2 1 mod n and g3 = gS3 1 mod n and stores secret values r; s as revocation key KRK.

All public cryptographic parameters q; p; n; g1; g2;
g3; h1; h2;Aproof are published and shared.

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 w1;w2 and makes the commitment: CCSP = hw11 hw22 mod p. U digitally signs CCSP , e.g. by RSA, and sends this signature SigU(CCSP ) with the construction of correctness proof PKfw1;w2 : CCSP = hw1 1 hw2 2 g to CSP, by notation of Camenisch and Stadler [20]. CSP checks the user's proof and the signature.

Then, CSP stores the pair CCSP ; SigU(CCSP ), signs the
commitment SigCSP (CCSP ) and sends it back to U.

Secondly, U requests a user master key from RM. U
computes A 0 proof = gw1 1 gw2 2 mod n and sends it with
CCSP ; SigCSP (CCSP ) and the construction of correctness
proof PKfw1;w2 : CCSP = hw1 1 hw2 2 ^ A0 proof = gw1
1 gw2 2 g to RM. RM checks the proof, CSP's signature SigCSP (CCSP ) and computes a secret contribution wRM such that Aproof = gw1 1 gw2 2 gwRM 3 mod n holds. After this step, U obtains own user master key KU which is triplet (w1;w2;wRM). U gets value wRM only with cooperation with RM which knows the factorization of n. To prevent the collusion attack, user's w1;w2 is not visible outwardly to a user because w1;w2 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 [23]. Further, U cannot make own user master key because only RM knows KRK. Any honest user can repeat the request for the user master key or demand other authentication proofs if CSP agrees with that.

3) Anonymous Access: In this phase, the i-th user Ui anonymously accesses Cloud Service Provider (CSP). This phase consists of two-messages used to authenticate Ui and establish a secret key between Ui and CSP.

_ Ui generates a random value random 2R f0; 1glsym.
The parameter lsym denotes the size of a shared secret
key for the symmetric cipher.

_ Ui encrypts random by the RSA public key of CSP.

_ The encrypted Enc PK server(random) is signed by the
Auth proof sign algorithm which ensures user anonymous
authentication. We assume that cryptographic parameters
such as (q; p; n; g1; g2; g3; h1; h2) and authentication
proof Aproof = gw11 gw2 2 gwRM 3 mod n are made
public and H is a secure hash function. To prove the
knowledge of the secret user key and sign random, Ui
performs the Auth proof sign algorithm:

$K_s$€R $\{0,1\}^l$
A = $A^{K}S_{proof}$ mod n
C1 = $g_3^{KSwRM}$ mod n
C2 = $g_3^{KS}$3 mod n
r1; r2 €R {0,1}gm+k+3l
r3 €R {0,1}gm+k+4:5l
rS €R{0,1}gm+k+l
Aproof = gr11 gr22 gr33 mod n
A_ = ArSproof mod n
C_1 = gr33 mod n
C_2 = grS3 mod n
c = H(Enc PK server(random);A;A_;Apr_oof ;C1;C2;
C_1;C_2)

z1 = r1 - cKSw1
z2 = r2 - cKSw2
z3 = r3 - cKSwRM
zS = rS - cKS

Finally, the signature elements A;A_;Apr_oof ;C1;C2;C_1; C_2; z1; z2; z3; zS, 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;A_;Apr_oof ;C1;C2;C_1;C_2; z1; z2; z3; zS. Then, CSP does the Auth proof verify algorithm:

$C_1 \underline{=} C_2^{rev}$ mod n
$A^{-}proof \underline{=} A^e g^{z1}{}_1\ g^{z2}{}_2\ g^{z3}{}_3$ mod n
$A \underline{=} A^e A^{zS}{}_{proof}$ mod n
$C^{-}1 \underline{=}$Ce1gz33 mod n
$C^{-}2 \underline{=}$Ce2gzS3 mod 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 _ K sym.
- CSP sends a response message (random_K sym) back to Ui.

4*) Secure Communication*: If the anonymous access phase is successful, the user Ui 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, Ui and CSP use the AES secret key K sym established in the previous phase.

*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 extract wRM. Firstly, RM extracts the random session value KS from C2 and the secret RM contribution value wRM from C1. Then, RM publishes wRM into a public blacklist. If the user uses revoked key then the equation C1 _ CwRM 2 mod 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 wRM from the suspected session received by CSP.

Then, RM finds the corresponding CCSP in the database. If CSP provides to RM the explicit evidence of user's breach, then RM sends CCSP to CSP. CSP is able to open the identity of a user from database but only with RM's help.

## V. 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.

TABLE II
PERFORMANCE EVALUATION OF OUR SOLUTION.

| Sessions | Sign/Authenticate Total time [ms] | Verify | Verify withrev=10 |
|---|---|---|---|
| 1 | 54 | 70 | 90 |
| 10 | 526 | 721 | 900 |
| 20 | 1042 | 1370 | 1712 |
| 50 | 2504 | 3328 | 4091 |

### A. 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 proofof 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, see Table II. In the Verify process, Table II 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 Fig. 2.

### B. Comparison with Related Work

We compare our Anonymous Access phase with the authentication phase of related solutions: Blantom solution [11], Luet al. solution [12] and Chow et al. solution [13]. 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 [12] takes 14 exp + 10 mul, Chow et al. solution [13] takes 14 exp + 15 mul and Blantom's solution [11] takes tens of pairing and exponentiation operations. The number of operations in Blantom's solution [11] 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 [12], Chow et al. solution [13] and Blantom solution [11] are pairing based and contain 6 pairing operations in the Verify process. Fig. 3 depicts the performance of the verify process 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 [12] or Chow et al. solution [13].

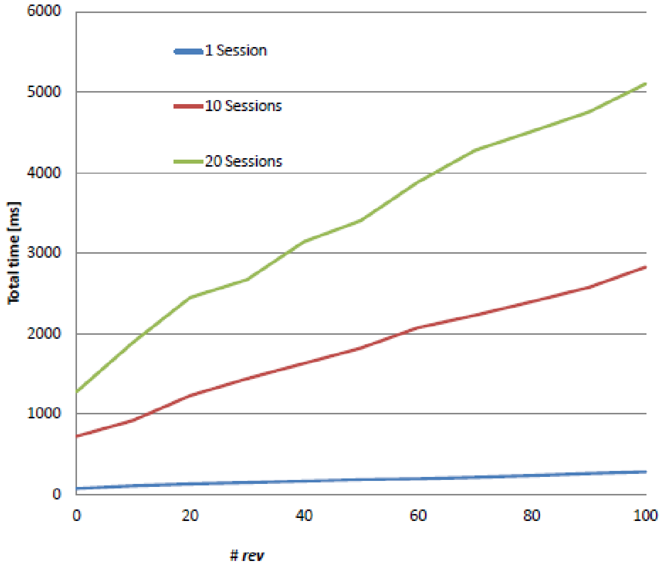


Fig. 2. Influence of the Length of the Blacklist on Total Time of Verification.

2. Influence of the Length of the Blacklist on Total Time of Verificat

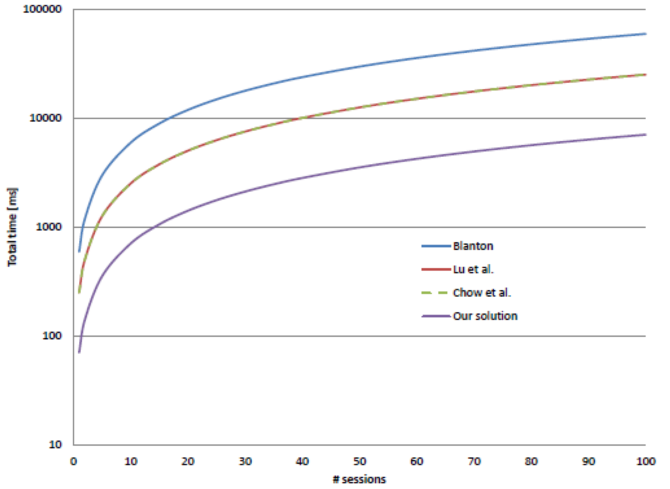


Fig. 3. Performance of the Verify Process.

## VI. CONCLUSION

The paper presents our novel security solution for privacypreserving cloud services. We propose non-bilinear group signatures to ensure anonymous authentication of cloud service clients. Our solution offers user anonymity in 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 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. Due to this fact, cloud service providers using our solution can authenticate more clients in the same time.

Our future plans are aimed on the modification of the revocation process. We would like to minimize the impact of the long-sized blacklist used in the Verify process. Also we will work on modification which cause that tamper resistant storage for user keys can be lack.

## 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. ACM, 2010, pp. 109–114.

[2] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-preserving public auditing for data storage security in cloud computing,” in INFOCOM, 2010 Proceedings IEEE, march 2010, pp. 1 –9.

[3] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, “Enabling public auditability and data dynamics for storage security in cloud computing,” Parallel and Distributed Systems, IEEE Transactions on, vol. 22, no. 5, pp. 847 –859, may 2011.

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

[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 middlewaRE, ser. COMSWARE ’09. New York, NY, USA: ACM, 2009, pp. 5:1–5:8. [Online]. Available: http://doi.acm.org/10.1145/1621890.1621897

[6] M. Jensen, S. Schage, and J. Schwenk, “Towards an anonymous access control and accountability scheme for cloud computing,” in Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, july 2010, pp. 540.

[7] D. Chaum and E. Van Heyst, “Group signatures,” in Advances in CryptologyEUROCRYPT91. Springer, 1991, pp. 257–265.

[8] P. Angin, B. Bhargava, R. Ranchal, N. Singh, M. Linderman, L. Othmane, and L. Lilien, “An entity-centric approach for privacy and identity management in cloud computing,” in Reliable Distributed Systems, 2010 29th IEEE Symposium on. IEEE, 2010, pp. 177–183.

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

[10] 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, NY, USA:
ACM, 2008, pp. 217–227.

[11] J. Camenisch and A. Lysyanskaya, “Signature schemes and anonymous credentials from bilinear maps.” in Advances in Cryptology – CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings, 2004, pp. 56–72.

[12] R. Lu, X. Lin, X. Liang, and X. S. Shen, “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, NY, USA: ACM, 2010, pp. 282–292. [Online]. Available: http://doi.acm.org/10.1145/1755688.1755723

[13] S. Chow, Y. He, L. Hui, and S. Yiu, “Spice–simple privacy-preserving identity-management for cloud environment,” in Applied Cryptography and Network Security. Springer, 2012, pp. 526–543.

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

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

[16] L. Malina and M. Zukal, “Secure authentication and key establishment in the sip architecture,” in Telecommunications and Signal Processing (TSP), 2011 34th International Conference on. IEEE, 2011, pp. 14–18.

[17] X. Boyen and B. Waters, “Full-domain subgroup hiding and constantsize group signatures,” Public Key Cryptography–PKC 2007, pp. 1–15,2007.

[18] 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.

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

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

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

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

[23] Z. Martinasek, T. Macha, O. Raso, J. Martinasek, and P. Silhavy, “Optimization of differential power analysis,”