SCIENCE CHINA
Information Sciences

• LETTER •

March 2020, Vol. 63 139107:1–139107:3
https://doi.org/10.1007/s11432-018-9706-0

General construction of revocable identity-based fully homomorphic signature

Congge XIE[1], Jian WENG[1*], Wei LU[2] & Lin HOU[1]

[1]*College of Information Science and Technology, Jinan University, Guangzhou* 510632*, China;*
[2]*School of Data and Computer Science, Sun Yat-sen University, Guangzhou* 510006*, China*

Received 8 May 2018/Revised 12 October 2018/Accepted 14 December 2018/Published online 11 February 2020

Citation Xie C G, Weng J, Lu W, et al. General construction of revocable identity-based fully homomorphic signature. Sci China Inf Sci, 2020, 63(3): 139107, https://doi.org/10.1007/s11432-018-9706-0

Dear editor,

Fully homomorphic signature (FHS) is a cryptographic primitive and allows anyone to conduct any computations over signatures and generate a signature which, when verified, matches the calculated value of the computational run on the messages. An identity-based fully homomorphic signature (IBFHS) extended from an FHS can avoid the cumbersome management of public-key certificates. In a revocable IBFHS scheme, an important issue that needs to be addressed is a user leaving an ID-based system. Taking a certified medical data analysis as an example, there are many different diseases research centers (such as cardiology, cancer and diabetes centers) demanding data sharing in a trusted National Health Center. Each research center runs a large-scale medical study on its respective field of disease. Cancer research center signs the collected data using their ID-based secret key and distributes the data to various cancer research groups and pharmaceutical companies for analysis. Some of these groups may have certain incentives to lie or miscalculate the outcomes of their analyses, and cannot which be trusted by the public. Even worse, the stakeholders in a cancer research center misbehave to achieve greater profits. Using revocable identity-based fully homomorphic signatures, not only can these groups publish their methodologies for the analysis and conduct any computations on the signed data as a short signature that certifies the correctness of the outcomes by the research center, the National Health Center can also provide a revocation mechanism to revoke the stakeholders in the cancer research center. Thus, it is important for an IBFHS to offer a revocation mechanism to revoke the associated public keys of misbehaving or malicious users. Thus far, there have been few revocable identity-based signature (RIBS) schemes [1,2] that resist quantum attacks from quantum computers. Unfortunately, they do not support full homomorphism and verifiable computations in untrusted cloud computing. Therefore, it is necessary and worthwhile to design a revocable identity-based fully homomorphic signature (RIBFHS) scheme.

In this study, we first describe studies related to threshold signatures [3], secret sharing [4], and universal computational extractor [5]. We then construct a (2, 2) vector secret sharing scheme over lattices, which is secure in the standard model under short integer solutions (SIS) assumption. Finally, we present a general construction of RIBFHS using the (2, 2) vector secret sharing and FHS as building blocks, which is EUF-CMA secure. Compared with the previous RIBS schemes [1, 2], our proposed RIBFHS scheme has two practical properties, namely, homomorphism and context-hiding, and its revocation mechanism not only makes the number of update keys be logarithmic in terms of the number of maximum users, but also indicates that the private key generator (PKG) does not need to encrypt/decrypt the pe-

* Corresponding author (email: cryptjweng@gmail.com)

riodic time update keys.

$(2,2)$ *vector secret sharing scheme.* We now construct a concrete (2, 2) vector secret sharing scheme from lattice as follows.

Initial phase. Choose a security parameter λ and conduct the following steps.

(1) Select the Gaussian parameter s_1 and set the parameters $n := n(\lambda)$, $q := q(\lambda)$, and $m := m(\lambda)$. These parameters are implicitly known to all of the algorithms below.

(2) Sample $(\boldsymbol{A}, \boldsymbol{T_A}) \leftarrow \mathrm{TrapGen}(1^n, 1^m, q)$, $(\boldsymbol{B}, \boldsymbol{T_B}) \leftarrow \mathrm{TrapGen}(1^n, 1^m, q)$ and two random matrices $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathbb{Z}_q^{n\times m}$.

(3) Compute $\boldsymbol{A}_{P_1,P_2} := (\boldsymbol{A}|\boldsymbol{A}_1 + H(P_1)\boldsymbol{B}|\boldsymbol{A}_2 + H(P_2)\boldsymbol{B})$ and run the algorithm $\boldsymbol{S} \leftarrow \mathrm{ExtBasis}(\boldsymbol{T_A}, \boldsymbol{A}_{P_1,P_2})$ with the hash function H.

(4) Output $\mathrm{pp} = (n, q, m, s_1, s_2)$, a participant set $P = \{P_1, P_2\}$, and the short basis of the secret $\boldsymbol{S}$ ($\Lambda_q^{\perp}(\boldsymbol{A}_{P_1,P_2})$'s short basis).

Sharing phase. This phase conducts the following steps.

(1) Choose a random matrix $\boldsymbol{U} \in \mathbb{Z}_q^{n\times m}$, and sample matrices $\boldsymbol{R}_1 \leftarrow \mathcal{D}_{m\times m}$ and $\boldsymbol{R}_2 \leftarrow \mathcal{D}_{m\times m}$. Run the algorithms $\boldsymbol{E}_{P_1} \leftarrow \mathrm{SamplePre}(\boldsymbol{A}, \boldsymbol{T_A}, -(\boldsymbol{A}_1 + H(P_1)\boldsymbol{B})\boldsymbol{R}_1, s_1)$, $\boldsymbol{E}_{P_2} \leftarrow \mathrm{SamplePre}(\boldsymbol{A}, \boldsymbol{T_A}, -(\boldsymbol{A}_2 + H(P_2)\boldsymbol{B})\boldsymbol{R}_2, s_1)$.

(2) Run the algorithm $\boldsymbol{E}_1 := \left(\begin{smallmatrix}\boldsymbol{E}_{1,1}\\ \boldsymbol{E}_{1,2}\end{smallmatrix}\right) \leftarrow \mathrm{SampleLeft}(\boldsymbol{A}, \boldsymbol{A}_1 + H(P_1)\boldsymbol{B}, \boldsymbol{T_A}, \boldsymbol{U}, s_1)$ and compute $\boldsymbol{H}_1 := \boldsymbol{E}_{1,2} - \boldsymbol{E}_{P_1}\boldsymbol{R}_1^{-1}\boldsymbol{E}_{1,1}$.

(3) Sample $\boldsymbol{E}_{2,2} \leftarrow (\mathcal{D}_{\mathbb{Z}^m}, s_1)^m$, compute $\boldsymbol{H}_2 := \boldsymbol{E}_{P_2}\boldsymbol{R}_2^{-1}\boldsymbol{E}_{2,2}$, and sample $\boldsymbol{E}_{2,1} \leftarrow \mathrm{SampleSubInv}(\boldsymbol{A}, \boldsymbol{H}_1 - \boldsymbol{H}_2, -\boldsymbol{U} - (\boldsymbol{A}_2 + H(P_2)\boldsymbol{B})\boldsymbol{E}_{2,2}, s_1)$.

(4) Set the sub-secret $\mathrm{sk}_1 := (\boldsymbol{E}_1, \boldsymbol{E}_{P_1}, \boldsymbol{R}_1)$ and sub-secret $\mathrm{sk}_2 := (\boldsymbol{E}_2, \boldsymbol{E}_{P_2}, \boldsymbol{R}_2)$, where $\boldsymbol{E}_2 := \left(\begin{smallmatrix}\boldsymbol{E}_{2,1}\\ \boldsymbol{E}_{2,2}\end{smallmatrix}\right)$.

(5) Return sk_1 and sk_2 to shareholders P_1 and P_2, respectively.

Construction phase. This phase proceeds as follows.

(1) Two shareholders set

$$\boldsymbol{S}_{P_1,P_2} = \begin{pmatrix} \boldsymbol{E}_{1,1} + \boldsymbol{E}_{2,1} & \boldsymbol{E}_{P_1} & \boldsymbol{E}_{P_2} \\ \boldsymbol{E}_{1,2} & \boldsymbol{R}_1 & \boldsymbol{0} \\ \boldsymbol{E}_{2,2} & \boldsymbol{0} & \boldsymbol{R}_2 \end{pmatrix}.$$

(2) Transform $\boldsymbol{S}_{P_1,P_2}$ into a $\boldsymbol{S}'$ by using Lemma 7.1 from [6] and output $\boldsymbol{S}'$.

Note that: $\boldsymbol{S}'$ is the short basis of $\Lambda_q^{\perp}(\boldsymbol{A}_{P_1,P_2})$ and is not necessarily $\boldsymbol{S}$.

For the details on this construction's proof, please refer to Appendix A.

With security parameter λ, we set the following:

$$n = \lambda,\ m = O(n\log q),\ \eta = \omega(\sqrt{\log m}),$$

$$s_1 \geqslant O(\sqrt{m}) \cdot \omega(\sqrt{\log m}),$$

$$\beta = \sqrt{3m} \cdot (1 + 2\eta\sqrt{m}),$$

$$q = \beta^{m/n} \cdot (1/\sqrt{m}).$$

General construction. Let VSS=(VSS.Initial, VSS.Sharing, VSS.Construction) be a (2, 2) vector secret sharing scheme, and FHS=(FHS.Setup, FHS.KeyGen, FHS.Sign, FHS.Eval, FHS.Verify) be a fully homomorphic signature scheme; BT is then a binary tree, and algorithm KUNodes comes from the scheme in [7]. Our general construction of the RIBFHS scheme is as follows.

Setup$(1^\lambda, 1^L, 1^l, N)$**.** The procedure uses a security parameter λ, a circuit depth L, a maximum size of the dataset l, and a maximal number of users N.

(1) Set $n =: n(\lambda, L)$, $q := q(n, L)$, $m := m(n, L)$, and $B = B(n, L) \in \mathbb{Z}$ as the upper bound of the signature size. These parameters are implicitly known to all of the algorithms below.

(2) The run initial phase of VSS is run to obtain the parameters pp_1 and master key msk.

(3) The algorithm $\mathrm{FHS.Setup}(1^\lambda, 1^L, 1^l, N)$ is run to obtain the other parameters pp_2.

(4) $\mathrm{PP} = (\mathrm{pp}_1, \mathrm{pp}_2)$ and the master secret key $\mathrm{MSK} = \mathrm{msk}$, a revocation list $\mathrm{RL} = \emptyset$, and the state $\mathrm{ST} = \mathrm{BT}$ are output.

PriKeyGen$(\mathrm{PP}, \mathrm{MSK}, \mathrm{id}, \mathrm{ST})$**.** Choose an unassigned leaf node ν in BT and allot it to id, and then conduct the steps below under identity $\mathrm{id} \in \mathcal{I}$ with PP, MSK, and the state ST.

(1) For any $\theta \in \mathrm{Path}(\nu)$, let identity id be the shareholder P_1 and run VSS.Sharing of VSS to obtain sk_θ.

(2) Output $\mathrm{SK}_{\mathrm{id}} := (\{\theta, \mathrm{sk}_\theta\}_{\theta\in\mathrm{Path}(\nu)}, \mathrm{ST})$.

KeyUpd$(\mathrm{PP}, \mathrm{MSK}, t, \mathrm{RL}, \mathrm{ST})$**.** The steps below are conducting during time $t \in \mathcal{T}$ with PP, MSK, revocation list RL, and state ST.

(1) $\forall\theta \in \mathrm{KUNodes}(\mathrm{BT}, \mathrm{RL}, t)$, let time t be the shareholder P_2 and run VSS.Sharing of VSS to obtain sk'_θ.

(2) Output $\mathrm{KU}_t := (\{\theta, \mathrm{sk}'_\theta\}_{\theta\in\mathrm{KUNodes}(\mathrm{BT},\mathrm{RL},t)})$.

Sign$(\mathrm{PP}, \mathrm{id}, t, \mathrm{SK}_{\mathrm{id}}, \mathrm{KU}_t, \tau, i, \mu)$**.** The steps below are conducted under id during time t with PP, its private key $\mathrm{SK}_{\mathrm{id}}$, update key KU_t, tag τ and its i-th a message μ.

(1) $\forall(i, \mathrm{sk}_i) \in \mathrm{SK}_{\mathrm{id}}$, $(j, \mathrm{sk}'_j) \in \mathrm{KU}_t$, if $\exists i = j$, then run VSS.Construction to obtain $\mathrm{SK}_{\mathrm{id},t}$.

(2) Run the algorithm of FHS,

$$\mathrm{FHS.Sign}(\mathrm{PP}, \mathrm{SK}_{\mathrm{id},t}, \tau, i, \mu)$$

to obtain a signature σ.

(3) Output σ.

Eval$(\mathrm{PP}, \mathrm{id}, t, \tau, \mu, \sigma, C)$**.** Output $\sigma_{|C|}$ using the evaluation algorithm of FHS, $\mathrm{FHS.Eval}(\mathrm{PP}, \tau, \mu, \sigma, C)$.

Verify$(\mathrm{PP}, \mathrm{id}, t, \tau, \mu, \sigma, C)$**.** Run the algorithm of FHS, FHS.Verify$(\mathrm{PP}, \mathrm{id}, t, \tau, \mu, \sigma, C)$ and output this result.

From the correctness of VSS, we know that the signing key $\mathrm{SK}_{\mathrm{id},t}$ generated from the algorithm SignKeyGen is the short basis of $\Lambda_q^{\perp}(\boldsymbol{A}_{\mathrm{id},t})$, which is a signing key that can be used in the algorithm FHS.Sign of FHS. Therefore, the general construction is correct by adding the correctness of the FHS scheme.

Our general RIBFHS scheme is EUF-sID-CMA secure, if the scheme VSS and the scheme FHS are secure. For the details on proof, please refer to Appendix B.

At last, for the comparisons between our general construction and existing lattice-based RIBS schemes, please refer to Appendix C.

Conclusion. We first proposed a (2, 2) vector secret sharing scheme over lattices, that is secure in the standard model under SIS assumption. Taking the (2, 2) vector secret sharing as a building block, we then presented a general construction of the RIBFHS scheme using an FHS scheme and proved its security. In the general construction, the number of update keys is logarithmic with the number of maximum users, and the PKG broadcasts the update keys regularly through public channels. Our proposed scheme also meets the homomorphism property and context-hiding property, which is suitable for use in a cloud computing environment.

Acknowledgements This work was supported by National Key R&D Plan of China (Grant No. 2017YFB0802203), National Natural Science Foundation of China (Grant Nos. U1736203, 61732021, 61472165, 61373158), Guangdong Provincial Engineering Technology Research Center on Network Security Detection and Defence (Grant No. 2014B090904067), Guangdong Provincial Special Funds for Applied Technology Research and Development and Transformation of Important Scientific and Technological Achieve (Grant No. 2016B010124009), Zhuhai Top Discipline-Information Security, Guangzhou Key Laboratory of Data Security and Privacy Preserving, Guangdong Key Laboratory of Data Security and Privacy Preserving, National Joint Engineering Research Center of Network Security Detection and Protection Technology.

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Xiang X Y. Adaptive secure revocable identity-based signature scheme over lattices. Comput Eng, 2015, 41: 126–129
2 Hung Y H, Tseng Y M, Huang S S. Revocable id-based signature with short size over lattices. Secur Commun Netw, 2017, 7571201: 1–9
3 Yang W J, Luo W Q, Luo X Z, et al. Fully distributed certificateless threshold signature without random oracles. Sci China Inf Sci, 2018, 61: 098101
4 Liu H, Li X H, Ma J F, et al. Reconstruction methodology for rational secret sharing based on mechanism design. Sci China Inf Sci, 2017, 60: 088101
5 Wang H G, Chen K F, Qin B D, et al. A new construction on randomized message-locked encryption in the standard model via UCEs. Sci China Inf Sci, 2017, 60: 052101
6 Micciancio D, Goldwasser S. Complexity of Lattice Problems: a Cryptographic Perspective. Berlin: Springer, 2012
7 Naor M, Nissim K. Certificate revocation and certificate update. IEEE J Sel Areas Commun, 2000, 18: 561–570