SCIENCE CHINA
Information Sciences

• HIGHLIGHT •

September 2018, Vol. 61 098101:1–098101:3
https://doi.org/10.1007/s11432-017-9244-9

Fully distributed certificateless threshold signature without random oracles

Wenjie YANG[1,2,3], Weiqi LUO[1*], Xizhao LUO[4*], Jian WENG[1,2,3] & Anjia YANG[1,2,3]

[1] *College of Cyber Security/College of Information Science and Technology, Jinan University, Guangzhou* 510632*, China;*
[2] *Guangdong Key Laboratory of Data Security and Privacy Preserving, Guangzhou* 511443*, China;*
[3] *Guangzhou Key Laboratory of Data Security and Privacy Preserving, Guangzhou* 511443*, China;*
[4] *School of Computer and Technology, Soochow University, Suzhou* 215006*, China*

Received 19 July 2017/Accepted 20 September 2017/Published online 19 April 2018

Citation 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(9): 098101, https://doi.org/10.1007/s11432-017-9244-9

A certificateless (t, n) threshold signature (CLTS) scheme allows at least t members to cooperatively sign a message on behalf of an n-member group, which is a good way to share the responsibility and authority. Many researchers have focused on CLTS schemes in the literature. However, these schemes numerously employ a single key generation center (KGC) and thus inevitably suffer from single point of failure and abuse of single key generator center. To settle these problems, Xiong et al. [1] introduced the concept of fully distributed CLTS, in which multiple distributed KGCs take the responsibility of generating and allocating the entity partial private key to distributed signers. Nevertheless, in their definition, each of the distributed KGCs directly generates and sends an entity partial private key share to a signer, which implicates two assumed conditions. One is that the number of KGCs and signers is the same, which results in poor scalability. The other one is that the collusion between KGCs and signers is forbidden, which weakens their security model.

In this article, we study the security concept and the design of fully distributed CLTS schemes. First, we refine fully distributed CLTS and enhance its security model. In our improved security model, we still consider two kinds of adversaries, a super public key replacement (PKR) attacker and a malicious-but-passive key generation center (MKGC) attacker. Besides the original attack abilities [1], these attackers are allowed to corrupt up to $t-1$ arbitrary signers and $k-1$ arbitrary MKGCs where t and k are the threshold values used to produce a completed signature and recover the system secret key, respectively. As a consequence, the two implicit assumptions in [1] are removed, which makes our security model more reasonable. Moreover, we give the first concrete fully distributed CLTS scheme by employing verifiable secret sharing [2] and distributed key generation (DKG) [3]. The security of the proposed scheme can be proven under the standard model like in [4, 5].

Construction. Let $(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, g)$ be a valid instance, where q denotes a big prime, $\mathbb{G}_1$ and $\mathbb{G}_2$ denote two q order cyclic groups, $\hat{e}$ denotes an admissible bilinear mapping $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ and g denotes a generator in $\mathbb{G}_1$. To design a more flexible scheme, collision-resistant hash functions, H_e: $\{0,1\}^* \to \{0,1\}^{n_e}$ and H_m: $\{0,1\}^* \times \mathrm{spk} \times \mathrm{PK} \to \{0,1\}^{n_m}$, are adopted to handle desired identities and messages. The proposed scheme consists of the following five algorithms: Setup, Extract-PartialPrivateKeyShare(ExtPPKS), SetUserKey, Sign, Verify.

Setup. All the KGCs first collectively pick two generators g and h of $\mathbb{G}_1$ where $\log_g h$ is unknown and then interactively execute Gennaro's

* Corresponding author (email: lwq@jnu.edu.cn, xzluo@suda.edu.cn)

© Science China Press and Springer-Verlag GmbH Germany 2018

DKG algorithm to generate partial implicit system secret keys (α_2, x', x_1, x_2, …, x_{n_e}, y', y_1, y_2, …, y_{n_m})$\in (\mathbb{Z}_p^*)^{n_e+n_m+3}$ and the corresponding system public keys $g_2 = g^{\alpha_2}$, $e' = g^{x'}$, $\boldsymbol{E} = \{e_i\}_{i=1}^{n_e} = \{g^{x_i}\}_{i=1}^{n_e}$, $w' = g^{y'}$, $\boldsymbol{W} = \{w_i\}_{i=1}^{n_m} = \{g^{y_i}\}_{i=1}^{n_m}$.

(1) To securely generate the system secret key $g_2^{\alpha_1}$, each KGC_i, $i = 1, 2, \ldots, m$, performs interactively as follows:

(a) Each KGC_i randomly chooses two $(k-1)$-degree polynomials $F_i(x)$ and $F_i'(x)$ over $\mathbb{Z}_p^*$: $F_i(x) = c_{i0} + c_{i1}x + \cdots + c_{ik-1}x^{k-1}$, $F_i'(x) = c_{i0}' + c_{i1}'x + \cdots + c_{ik-1}'x^{k-1}$. Let $P_{i0} = c_{i0} = F_i(0)$. KGC_i opens $T_{il} = g^{c_{il}}h^{c_{il}'} \mod p$ for $l = 0, 1, \ldots, k-1$. KGC_i computes the shares $P_{ij} = F_i(j)$, $P_{ij}' = F_i'(j) \mod p$ for $j = 1, 2, \ldots, m$ and sends P_{ij}, P_{ij}' to KGC_j.

(b) KGC_j checks the shares from the other KGC_i by verifying the equality $g^{P_{ij}}h^{P_{ij}'} = \prod_{l=0}^{k-1}(T_{il})^{j^l} \mod p$. If the equality does not hold for an index i, then KGC_j opens a complaint to KGC_i.

(c) Each KGC_i opens P_{ij} and P_{ij}' if it receives a complaint from KGC_j. Otherwise, KGC_i is disqualified.

(d) KGC_j marks as disqualified any KGC_i that either received more than $k-1$ complaints in Step (1b), or answered to a complaint in Step (1c) with invalid values.

(e) Each KGC_i owns the same set of non-disqualified $\mathrm{QUAL_{KGC}}$ and recovers their secret share $\alpha_{1i} = \sum_{j\in\mathrm{QUAL_{KGC}}} P_{ji} \mod p$ and $r_i = \sum_{j\in\mathrm{QUAL_{KGC}}} P_{ji}' \mod p$. Note that, the completed system secret key $g_2^{\alpha_1}$ is not explicitly obtained by anyone, but it equals $\mathrm{ssk} = \prod_{i\in\mathrm{QUAL_{KGC}}} g_2^{P_{i0}} \mod p$.

(2) Each KGC_i jointly generates $g_1 = g^{\alpha_1}$ and $\mathrm{fvk}_i = g^{\alpha_{1i}}$ for $i \in \mathrm{QUAL_{KGC}}$:

(a) Each KGC_i opens $A_{il} = g^{c_{il}}$, $l = 0, \ldots, k-1$.

(b) KGC_j verifies the values received from KGC_i by checking $g^{P_{ij}} = \prod_{l=0}^{k-1}(A_{il})^{j^l}$ for $i \in \mathrm{QAUL_{KGC}}$. If the equality does not hold, KGC_j complains against KGC_i by opening the values P_{ij}, P_{ij}'.

(c) KGC_i in $\mathrm{QUAL_{KGC}}$ can compute and publish public parameters $g_1 = \prod_{j\in\mathrm{QUAL_{KGC}}} A_{j0}$ and $\mathrm{fvk}_i = g^{\alpha_{1i}} = \prod_{j\in\mathrm{QUAL_{KGC}}}\prod_{l=0}^{k-1}(A_{jl})^{i^l}$.

At last, the system public keys are $\mathrm{spk} = (g, h, g_1, g_2, e', \boldsymbol{E}, w', \boldsymbol{W})$.

ExtPPKS. Let e be an entity and set $E = H_e(e)$. Set $\mathscr{E}$ to be the set of indices $i \subset \{1, 2, \ldots, n_e\}$ where $E[i] = 1$. In order to generate an entity partial private key $D_e = (D_{e1}, D_{e2}) = (g_2^{\alpha_1}(e'\prod_{i\in\mathscr{E}} e_i)^{r_e}, g^{r_e})$, each KGC_i performs respectively as follows:

(a) Choose $b_{i1}, b_{i2}, \ldots, b_{it-1}$, $b_{i0}', b_{i1}', \ldots, b_{it-1}'$ from $\mathbb{Z}_p^*$ uniformly at random and define $g_i(x) = b_{i0} + b_{i1}x + \cdots + b_{it-1}x^{t-1}$, $g_i'(x) = b_{i0} + b_{i1}x + \cdots + b_{it-1}x^{t-1}$ where $b_{i0} = \lambda_i\alpha_{1i}$. Note that, $\lambda_1, \lambda_2, \ldots, \lambda_m$ are the Lagrange coefficients.

(b) Compute and publish $E_{il} = E(b_{il}, b_{il}') = \hat{e}(g, g_2)^{b_{il}}\hat{e}(g, h)^{b_{il}'}$ for $l = 0, 1, \ldots, t-1$ as the commitments of b_{i0} and $g_i(x)$.

(c) Pick $r_{eij}' \in \mathbb{Z}_q^*$ at random and compute $D_{eij1} = g_2^{g_i(j)}(e'\prod_{j\in\mathscr{E}} e_j)^{r_{eij}'}$, $D_{eij2} = g^{r_{eij}'}$, $D_{eij3} = g_i'(j)$, for $j = 1, 2, \ldots, n$.

At last, set the first verification key share $\mathrm{fvk}_{ij} = \hat{e}(g, g_2)^{g_i(j)}$ and secretly send (D_{eij1}, D_{eij2}, D_{eij3}) to the signer $\mathcal{S}_j$.

SetUserKey. All of the signers first collectively pick a generator h' of $\mathbb{G}_1$ where $\log_g h'$ is unknown and then interactively execute Gennaro's DKG algorithm to generate partial implicit secret values $(\beta_{e2}, z_e', z_{e1}, z_{e2}, \ldots, z_{en_m}) \in \mathbb{Z}_p^*$ and the corresponding public keys $(g_{e2} = g^{\beta_{e2}}, v_e' = g^{z_e'}, v_{e1} = g^{z_{e1}}, v_{e2} = g^{z_{e2}}, \ldots, v_{en_m} = g^{z_{en_m}}) \in \mathbb{G}_1$.

(1) To obtain the implicit secret value $\mathrm{SV}_e = g_{e2}^{\beta_{e1}} \in \mathbb{G}_1$, each $\mathcal{S}_i$ performs interactively as follows:

(a) $\mathcal{S}_i$ picks two random $(t-1)$-degree polynomials $G_i(x)$ and $G_i'(x)$ over $\mathbb{Z}_p^*$: $G_i(x) = c_{i0} + c_{i1}x + \cdots + c_{it-1}x^{t-1}$, $G_i'(x) = c_{i0}' + c_{i1}'x + \cdots + c_{it-1}'x^{t-1}$. Let $P_{i0} = c_{i0} = G_i(0)$. $\mathcal{S}_i$ opens $C_{il} = g^{c_{il}}h'^{c_{il}'} \mod p$ for $l = 0, 1, \ldots, t-1$. $\mathcal{S}_i$ produces and sends the shares $P_{ij} = G_i(j)$, $P_{ij}' = G_i'(j) \mod p$ for $j = 1, 2, \ldots, n$ to $\mathcal{S}_j$.

(b) $\mathcal{S}_j$ checks the shares from any other signer $\mathcal{S}_i$ by verifying $g^{P_{ij}}h'^{P_{ij}'} = \prod_{l=0}^{t-1}(C_{il})^{j^l} \mod p$. If the equality does not hold for an index i, then $\mathcal{S}_j$ opens a complaint to $\mathcal{S}_i$.

(c) $\mathcal{S}_i$ opens P_{ij} and P_{ij}', if he receives a complaint from $\mathcal{S}_j$. Otherwise, $\mathcal{S}_i$ is disqualified.

(d) $\mathcal{S}_j$ marks as disqualified any other signer $\mathcal{S}_i$ that either received more than $t-1$ complaints in Step (1b), or answered to a complaint in Step (1c) with invalid values.

(e) $\mathcal{S}_i$ owns the same non-disqualified set QUAL_e and recovers their secret share $\beta_{e1i} = \sum_{j\in\mathrm{QUAL}_e} P_{ji} \mod p$ and $\beta_{e1i}' = \sum_{j\in\mathrm{QUAL}_e} P_{ji}' \mod p$. Note that, the completed secret value $g_{e2}^{\beta_{e1}}$ is not explicitly obtained by anyone, but it equals $\mathrm{SV}_e = \prod_{i\in\mathrm{QUAL}_e} g_{e2}^{P_{i0}} \mod p$.

(2) Each $\mathcal{S}_i$ jointly generates the public key $g_{e1} = g^{\beta_{e1}}$ and the second verification key share $\mathrm{svk}_i = \hat{e}(g, g_{e2})^{\beta_{e1i}}$ for $i \in \mathrm{QUAL}_e$:

(a) $\mathcal{S}_i$ broadcasts $A_{il} = g^{a_{il}}$ for $l = 0, \ldots, t-1$.

(b) $\mathcal{S}_j$ checks the share P_{ij} from the other signers $\mathcal{S}_j$ by verifying $g^{P_{ij}} = \prod_{l=0}^{t-1}(A_{il})^{j^l}$ for

$i \in \mathrm{QUAL}_e$. If the equality fails for an index i, $\mathcal{S}_j$ complains against $\mathcal{S}_i$ by broadcasting P_{ij} and P'_{ij}.

(c) $\mathcal{S}_i$ in QUAL_e computes and publishes $g_{e1} = \prod_{j\in\mathrm{QUAL}_e} A_{j0}$ and $\mathrm{svk}_{ei} = \hat{e}(g, g_{e2})^{\beta_{e1i}} = \hat{e}(g_{e2}, \prod_{j\in\mathrm{QUAL}_e} \prod_{l=0}^{t-1} (A_{jl})^{i^l})$.

When receiving the partial private key share $(D_{eij1}, D_{eij2}, D_{eij3})$ from KGC_i, $\mathcal{S}_j$ verifies the following equations $\hat{e}(D_{eij1}, g) \stackrel{?}{=} \mathrm{fvk}_{ij} \cdot \hat{e}(e' \prod_{i\in\mathscr{E}} e_i, D_{eij2})$ and $\hat{e}(D_{eij1}, g)\hat{e}(g, h)^{D_{eij3}} \stackrel{?}{=} \hat{e}(e' \prod_{k\in\mathscr{E}} e_k, D_{eij2}) \prod_{l=1}^{t_1} E_{il}^{j^l}$. If the verifications fail, the corresponding share $(D_{eij1}, D_{eij2}, D_{eij3})$ assigned to $\mathcal{S}_j$ is invalid. Otherwise, $\mathcal{S}_j$ computes $D_{ej1} = \prod_{i=1}^{k} D_{eij1}$, $D_{ej2} = \prod_{i=1}^{k} D_{eij2}$ and $\mathrm{fvk}_j = \prod_{i=1}^{k} \mathrm{fvk}_{ij}$ where without loss of generality, we assume that the first k KGC_i in $\mathrm{QUAL}_{\mathrm{KGC}}$ will be used to generate the partial private key shares. At last, $\mathcal{S}_j$ sets $\mathrm{PK}_e = \{g_{e1}, g_{e2}, v'_e, \boldsymbol{V}_e\}$ and $\mathrm{SK}_{ej} = (\mathrm{SV}_{ej}, D_{ej1}, D_{ej2})$.

Sign. Given a message M, the system public key spk and an entity e with the public key PK_e, compute $N = H_m(M\|\mathrm{spk}\|\mathrm{PK}_e)$ and set $\mathscr{M}$ to be the set of indices $j \subset \{1, 2, \ldots, n_m\}$ where $N[j] = 1$. Next, each $\mathcal{S}_j$ first picks r_{mj} from $\mathbb{Z}_p^*$ and then computes $D_w = w' \prod_{l\in\mathscr{M}} w_l$ and $D_v = v' \prod_{l\in\mathscr{M}} v_l$, finally broadcasts

$$\begin{aligned}\sigma_j &= (\sigma_{j1}, \sigma_{j2}, \sigma_{j3}, \sigma_{j4})\\ &= (\mathrm{SV}_{ej} D_v^{r_{mj}}, D_{ej1} D_w^{r_{mj}}, D_{ej2}, g^{r_{mj}}).\end{aligned}$$

On input σ_j, fvk_j and svk_j, any participant checks the following equations:

$$\begin{aligned}&\hat{e}(\sigma_{j1}, g) \stackrel{?}{=} \mathrm{svk}_j \cdot \hat{e}(D_v, \sigma_{j4}),\\ &\hat{e}(\sigma_{j2}, g) \stackrel{?}{=} \mathrm{fvk}_j \cdot \hat{e}(D_u, \sigma_{j3})\hat{e}(D_w, \sigma_{j4}).\end{aligned}$$

If the check fails, the participant broadcasts a complaint against $\mathcal{S}_j$.

Let $\lambda_1, \lambda_2, \ldots, \lambda_t$ be the Lagrange coefficients. Without loss of generality, we assume that the first t signers jointly recover σ of M on e as follows:

$$\begin{aligned}\sigma &= (\sigma_1, \sigma_2, \sigma_3, \sigma_4)\\ &= \left(\prod_{j=1}^{t} \sigma_{j1}^{\lambda_j}, \prod_{j=1}^{t} \sigma_{j2}^{\lambda_j}, \prod_{j=1}^{t} \sigma_{j3}^{\lambda_j}, \prod_{j=1}^{t} \sigma_{j4}^{\lambda_j}\right).\end{aligned}$$

Verify. Given σ on M of e with $\mathrm{PK}_e = \{g_{e1}, g_{e2}, v'_e, \boldsymbol{V}_e\}$, the verifier first computes $D_u = e' \prod_{i\in\mathscr{E}} e_i$, $D_w = w' \prod_{j\in\mathscr{M}} w_j$ and $D_v = v' \prod_{j\in\mathscr{M}} v_j$, then checks the following equalities:

$$\begin{aligned}&\hat{e}(\sigma_1, g) \stackrel{?}{=} \hat{e}(g_{e1}, g_{e2})\hat{e}(D_v, \sigma_4),\\ &\hat{e}(\sigma_2, g) \stackrel{?}{=} \hat{e}(g_1, g_2)\hat{e}(D_u, \sigma_3)\hat{e}(D_w, \sigma_4).\end{aligned}$$

Output 1 if it is valid. Otherwise, output 0.

Conclusion. In this article, we first refined the fully distributed CLTS definition and then improved its security model. Finally, we gave the first fully distributed CLTS scheme provably secure in the standard model. Further discussion are available in the supporting information.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61373158, 61472165, 61732021, 61702222), 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), China Postdoctoral Science Foundation Funded Project (Grant No. 2017M612842), Zhuhai Top Discipline–Information Security.

Supporting information Appendixes A–F. 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 Xiong H, Li F G, Qin Z G. Certificateless threshold signature scheme provably secure in the standard model. Inf Sci, 2013, 237: 73–81
2 Zhang F T, Zhang J. Efficient and information theoretical secure verifiable secret sharing over bilinear groups. Chinese J Electron, 2014, 23: 13–17
3 Gennaro R, Jarecki S, Krawczyk H, et al. Secure distributed key generation for discrete-log based cryptosystems. J Cryptol, 2007, 20: 51–83
4 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
5 Wei F S, Ma J F, Zhang R J, et al. An efficient and practical threshold gateway oriented password-authenticated key exchange protocol in the standard model. Sci China Inf Sci, 2017, 60: 072103