SCIENCE CHINA
Information Sciences

• LETTER •

August 2020, Vol. 63 189101:1–189101:3
https://doi.org/10.1007/s11432-018-9589-x

Secure two-party SM9 signing

Yongheng MU[1,2,3], Haixia XU[1,2,3*], Peili LI[1,2] & Tianjun MA[1,2,3]

[1]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093*, China;*
[2]*Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing* 100093*, China;*
[3]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing* 100049*, China*

Received 1 June 2018/Accepted 30 August 2018/Published online 26 March 2020

Citation Mu Y H, Xu H X, Li P L, et al. Secure two-party SM9 signing. Sci China Inf Sci, 2020, 63(8): 189101, https://doi.org/10.1007/s11432-018-9589-x

Dear editor,

With the rapid development of digital currencies and blockchain technologies, an increasing number of countries are planning to launch or have already launched their own central bank-issued digital currency (CBDC). In addition to drawing on advanced technologies such as blockchain, the ideal CBDC should also reconcile the functional conflicts between the requirements of convenience and security, and address the technical conflicts among the requirements of privacy (such as confidentiality of financial intelligence), audit, supervision, tracking and cracking down on illegal acts. However, clearly, anonymity is an important reason for the popularity of digital currencies represented by Bitcoin [1], and it is achieved through the elliptic curve digital signature algorithm (ECDSA) embedded in the system. The public and private key pairs used in the ECDSA have no association with the user's identity in real life. Thus, in the CBDC system, it is necessary to embed an identity-based digital signature, so that the government can associate the users' identities with their addresses (public keys) in the blockchain network when necessary.

Shamir [2] introduced the concept of identity-based cryptography in 1984, in which the user's private key is generated by a key generation center (KGC) based on master keys and the user ID, and the user's public key is uniquely identified by its own ID. Subsequently, Sakai et al. [3] and Boneh and Franklin [4] proposed different identity-based cryptographic schemes. In 2016, the State Cryptography Administration of China released the identity-based cryptography algorithm SM9, including a digital signature algorithm, key exchange protocol, key encapsulation mechanism, and public key encryption algorithm. It is worth mentioning that, in 2017, the digital signature algorithm of SM9 (SM9-DSA) was unanimously adopted as an international standard at the ISO/IEC JTC1 SC27 working group meeting.

While SM9-DSA can be considered as a candidate for the identity-based digital signature used in the CBDC system, there is another problem in that key protection of the digital wallet needs to be handled. Unlike traditional banking transactions, digital "coin" transactions are authorized only with a digital signature generated by the user's private key. Thus, one's digital "coins" are only as secure as the private key that can authorize their transfers; if the private key is compromised, the "coins" are certainly stolen. Furthermore, once a digital "coin" transaction is enacted, it is irreversible because of the anti-tamper property of the blockchain, even though the "coins" are known to have been stolen. Recently, in 2017, Lindell [5] proposed a fast secure two-party signing protocol based on the ECDSA to protect the private key of Bitcoin's wallet. In fact, two-party signing can be viewed as a special case of (t, n)-threshold signature algorithms [6, 7], where $t = 2$, $n = 2$.

We present a construction of the two-party sign-

* Corresponding author (email: xuhaixia@iie.ac.cn)

ing protocol based on the identity-based SM9-DSA by using the additive homomorphic property of the Paillier encryption scheme. In the construction, to combine SM9-DSA and the Paillier encryption scheme, we introduce a random technique to avoid the Paillier encryption of a point over an elliptic curve that the SM9-DSA is based on.

Technical concepts. Assume that the two-party protocol is designed to generate user A's (with identity ID_A) signature of message M. Let A_1 and A_2 denote the two parties participating in the protocol. It is easy to directly apply the method proposed by Lindell [5]. However, we find that it is not feasible for the strategy of A_1 with respect to the security proof. The primary cause is that the private key in SM9-DSA is a point over an elliptic curve, which makes the proof process either unworkable or contrary to the underlying hard problem, namely the elliptic curve discrete logarithm problem (ECDLP). We solve this by changing the holding order of the shared private key. That is, let the share that A_1 holds be a random number, and the share that A_2 holds be the relative point over the elliptic curve, such that the product of the two shares equals the original private key. However, this method of key sharing will pose a new problem for A_2. Paillier encryption [8] is only for integers rather than a point over an elliptic curve. We solve this by introducing a new random number, which avoids the Paillier encryption of a point over an elliptic curve, while maintaining its confidentiality.

As in Lindell's protocol [5], we use the additive homomorphic properties of Paillier encryption (assume that the public/private key pair is $(\mathrm{pk}, \mathrm{sk})$, where $\mathrm{pk} = n = p_1p_2$; p_1, p_2 are two random large prime numbers chosen independently). Moreover, our protocol is presented in the $\mathcal{F}_{\mathrm{zk}}$ and $\mathcal{F}_{\text{com-zk}}$ hybrid model [9]. Specifically, there are three relations that must be proved through ideal zero-knowledge (ZK) functionalities, denoted by $\mathcal{F}_{\mathrm{zk}}^{R_{\mathrm{PE}}}$, $\mathcal{F}_{\mathrm{zk}}^{R_{\mathrm{DL}}}$, and $\mathcal{F}_{\mathrm{zk}}^{R_{\mathrm{PEDL}}}$, respectively. For the sake of simplicity, we list these relations without the proof.

(1) Correctness of the generation of Paillier public keys $R_{\mathrm{PE}} = \{(n, (p_1, p_2)) | n = p_1p_2 \text{ and } p_1, p_2 \text{ are prime numbers}\}$.

(2) Knowledge of the discrete log of an integer $R_{\mathrm{DL}} = \{(G_T, g, N, g_1, r_1) | g_1 = g^{r_1}\}$.

(3) A Paillier ciphertext encrypted by a discrete log $R_{\mathrm{PEDL}} = \{((c, \mathrm{pk}, g_1, G, g, N), (r_1, \mathrm{sk})) | r_1 = \mathrm{Dec}_{\mathrm{sk}}(c) \text{ and } g_1 = g^{r_1} \text{ and } r_1 \in [1, N-1]\}$, where pk is a given Paillier public key and sk is the corresponding private key.

Our protocol. As in the original SM9-DSA, the system parameters are defined as follows. Let the finite field F_q be the base field of an elliptic curve, and a, b be the parameters of the elliptic curve equation. Let N be the prime factor of the order of the elliptic curve $E(F_q)$, and cf be the residual factor relative to N. Let k be the embedding degree (relative to N) of the elliptic curve $E(F_q)$. Let G_1 be a cyclic subgroup of order N of the elliptic curve $E(F_{q^{d_1}})$ (where d_1 divides k) with base point (generator) P_1; G_2 be another cyclic subgroup of order N of the elliptic curve $E(F_{q^{d_2}})$ (where d_2 divides k) with base point (generator) P_2; and G_T be a multiplicative cyclic subgroup of order N of the finite field $F_{q^k}^*$.

The sharing of user A's private signing key ds_A is completed by the KGC itself. Similar to the original SM9-DSA, we only use multiplicative sharing of private key ds_A at the final step. Because only addition and scalar multiplication operations can be performed on the elliptic curve, it is impossible for the KGC to generate two points such that their product is ds_A. Thus, the KGC chooses a random number $a \in [1, N-1]$, computes $\mathrm{ds}_{A_1} = a^{-1} \bmod N$ as A_1's private key, and computes $\mathrm{ds}_{A_2} = a \cdot \mathrm{ds}_A$ as A_2's private key, such that $\mathrm{ds}_A = \mathrm{ds}_{A_1} \cdot \mathrm{ds}_{A_2}$. The two-party key sharing is defined as follows.

(1) The KGC chooses a random $\mathrm{ks} \in [1, N-1]$ as the master private key and computes an element $P_{\text{pub-s}} = \mathrm{ks} \cdot P_2$ of group G_2 as the master public key.

(2) The KGC chooses and publicizes the identifier hid for the signing private key generation function denoted by one byte.

(3) The KGC computes $t_1 = H_1(\mathrm{ID}_A || \mathrm{hid}, N) + \mathrm{ks}$ in the finite field F_N. If $t_1 = 0$, then it reselects the master private key, computes and publicizes the master public key, and updates all registered users' signing private keys; else, it computes $t_2 = \mathrm{ks} \cdot t_1^{-1}$.

(4) The KGC chooses a random $a \in [1, N-1]$, computes $\mathrm{ds}_{A_1} = a^{-1} \bmod N$, and sends it to A_1; it computes $\mathrm{ds}_{A_2} = at_2 \cdot P_1$ and sends it to A_2.

Now, both A_1 and A_2 have the shared private keys ds_{A_1} and ds_{A_2}, respectively. The two-party signing protocol is described as follows. First, the two parties cooperatively generate a random value ω that will be used to generate the signature, by running the Diffie-Hellman exchange protocol. Then, A_1 generates a Paillier encryption scheme and encrypts r_1 using the Paillier public key pk. Upon receiving $c_1 = \mathrm{Enc}_{\mathrm{pk}}(r_1)$, A_2 needs to choose another random $r_3 \in [1, N-1]$, and compute $c_3 = \mathrm{Enc}_{\mathrm{pk}}(r_3(r_1r_2 - h))$ (using the homomorphic property of the Paillier encryption scheme) and $c_4 = r_3^{-1} \cdot \mathrm{ds}_{A_2}$. Upon receiving c_2, c_3, and c_4 from A_2, party A_1 can compute

$s_1 = \mathrm{Dec_{sk}}(c_3) = \mathrm{Dec_{sk}Enc_{pk}}(r_3(r_1r_2 - h)) = r_3(r_1r_2 - h)$, and then compute $S = s_1c_4\mathrm{ds}_{A_1} = r_3(r_1r_2-h)r_3^{-1}\mathrm{ds}_{A_2}\mathrm{ds}_{A_1} = (r_1r_2-h)\mathrm{ds}_{A_2}\mathrm{ds}_{A_1}$. To prevent the case that A_2 is corrupted and sends something wrong, A_1 needs to verify that the signature is valid before outputting it.

The formal definition of the two-party signing protocol is presented as follows.

Inputs.

(1) Party A_1 has ds_{A_1} received from the KGC, the message M, and a unique session id sid.

(2) Party A_2 has ds_{A_2} received from the KGC, the message M, and a unique session id sid.

(3) A_1 and A_2 locally verify that sid has not been used before (if it has been, the protocol is not executed).

The protocol.

• **A_1's first message:**

(1) A_1 computes the element $g = e(P_1, P_{\text{pub-s}})$ of group G_T.

(2) A_1 chooses a random $r_1 \in [1, N-1]$, and computes an element $g_1 = g^{r_1}$ of group G_T.

(3) A_1 sends $(\text{com-prove}, \text{sid}||1, g_1, r_1)$ to $\mathcal{F}_{\text{com-zk}}^{R_{\text{DL}}}$ (i.e., A_1 sends a commitment to g_1 and a ZK proof of knowledge of its discrete log).

• **A_2's first message:**

(1) A_2 receives $(\text{proof-receipt}, \text{sid}||1)$ from $\mathcal{F}_{\text{com-zk}}^{R_{\text{DL}}}$.

(2) A_2 computes the element $g = e(P_1, P_{\text{pub-s}})$ of group G_T.

(3) A_2 chooses a random $r_2 \in [1, N-1]$, and computes the element $g_2 = g^{r_2}$ of group G_T.

(4) A_2 sends $(\text{prove}, \text{sid}||2, g_2, r_2)$ to $\mathcal{F}_{\text{zk}}^{R_{\text{DL}}}$.

• **A_1's second message:**

(1) A_1 receives $(\text{proof}, \text{sid}||2, g_2)$ from $\mathcal{F}_{\text{zk}}^{R_{\text{DL}}}$. If not, it aborts.

(2) A_1 sends $(\text{decom-proof}, \text{sid}||1)$ to $\mathcal{F}_{\text{com-zk}}^{R_{\text{DL}}}$.

(3) A_1 generates a Paillier public/private key pair (pk, sk) and computes $c_1 = \mathrm{Enc_{pk}}(r_1)$.

(4) A_1 sends $(\text{prove}, \text{sid}||1, n, (p_1, p_2))$ to $\mathcal{F}_{\text{zk}}^{R_{\text{PE}}}$, where $\text{pk} = n = p_1p_2$.

(5) A_1 sends $(\text{prove}, \text{sid}||1, (c_1, \text{pk}, g_1), (r_1, \text{sk}))$ to $\mathcal{F}_{\text{zk}}^{R_{\text{PEDL}}}$.

• **A_2's second message:**

(1) A_2 receives $(\text{decom-proof}, \text{sid}||1, g_1)$ from $\mathcal{F}_{\text{com-zk}}R_{\text{DL}}$, $(\text{proof}, \text{sid}||1, n)$ from $\mathcal{F}_{\text{zk}}^{R_{\text{PE}}}$, and $(\text{proof}, \text{sid}||1, (c_1, \text{pk}, g_1))$ from $\mathcal{F}_{\text{zk}}^{R_{\text{PEDL}}}$; if not, it aborts.

(2) A_2 computes the element $\omega = (g_1)^{r_2} = g^{r_1r_2}$ of group G_T.

(3) A_2 computes the integer $h = H_2(M||\omega, N)$.

(4) A_2 chooses a random $r_3 \in [1, N-1]$, computes $c_2 = \mathrm{Enc_{pk}}(h)$, $c_3 = \mathrm{Enc_{pk}}(r_3(r_1r_2 - h))$, $c_4 = r_3^{-1} \cdot \mathrm{ds}_{A_2}$, and sends c_3 and c_4 to A_1.

• **A_1 generates output:**

(1) A_1 computes the element $\omega = (g_2)^{r_1} = g^{r_2r_1}$ of group G_T, and the integer $h = H_2(M||\omega, N)$.

(2) A_1 computes $s_1 = \mathrm{Dec_{sk}}(c_3) = \mathrm{Dec_{sk}Enc_{pk}}(r_3(r_1r_2 - h)) = r_3(r_1r_2 - h)$, and then computes $S = s_1c_4\mathrm{ds}_{A_1} = r_3(r_1r_2 - h)r_3^{-1}\mathrm{ds}_{A_2}\mathrm{ds}_{A_1} = (r_1r_2 - h)\mathrm{ds}_{A_2}\mathrm{ds}_{A_1}$.

(3) A_1 verifies that (h, S) is a valid signature with a public key identified by the user A's ID ID_A. If yes, it outputs the signature (h, S); otherwise, it aborts.

If any party aborts at any point, then it does not participate in any future signing protocol execution.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2017YFB0802500).

Supporting information The security proof of the protocol. 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 Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. 2009. http://www.bitcoin.org/bitcoin.pdf
2 Shamir A. Identity-based cryptosystems and signature schemes. In: Proceedings of Workshop on the Theory and Application of Cryptographic Techniques, 1984. 47–53
3 Sakai R, Ohgishi K, Kasahara M. Cryptosystems based on pairing. In: Proceedings of Symposium on Cryptography and Information Security, 2000. 26–28
4 Boneh D, Franklin M. Identity-based encryption from the Weil pairing. In: Proceedings of Annual International Cryptology Conference, 2001. 213–229
5 Lindell Y. Fast secure two-party ECDSA signing. In: Proceedings of Annual International Cryptology Conference, 2017. 613–644
6 Desmedt Y, Frankel Y. Threshold cryptosystems. In: Proceedings of Conference on the Theory and Application of Cryptology, 1989. 307–315
7 Desmedt Y, Frankel Y. Shared generation of authenticators and signatures. In: Proceedings of Annual International Cryptology Conference, 1991. 457–469
8 Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques, 1999. 223–238
9 Canetti R, Lindell Y, Ostrovsky R, et al. Universally composable two-party and multi-party secure computation. In: Proceedings of the 34th Annual ACM Symposium on Theory of Computing, 2002. 494–503