1 The security proof of the protocol

1.1 Security in the hybrid model

The $\mathcal{F}$-hybrid model is informally described as [1]: consider a protocol π that operates in a hybrid model of computation where parties can communicate as usual, and in addition have ideal access to an unbounded number of copies of some ideal functionalities $\mathcal{F}$. According to the model, the communication channels are public, which means that the adversary can see all the message sent, and deliver or block these messages at will. In the protocol, we assume that the adversarys computational power is P.P.T.. We will prove the security of our two-party signing protocol in the $\mathcal{F}_{zk}$ and $\mathcal{F}_{com-zk}$ hybrid model. According to the contributions of [2] and [3], we can conclude that working in this model is soundness.

1.2 Proof of Security

As stated in [4], there are two approaches for capturing the security properties of signature algorithms. The first one is, in [5], they proposed a widely accepted security requirement of a digital signature algorithm, called existential unforgeability against chosen message attacks. Specifically, the requirement is that, if both the public and private keys are generated honestly, then the signatures generated honestly also will always pass the verification algorithm; and in addition, it will be infeasible for an adversary to produce a new signature, i.e., a message m that has not been signed before, and an alleged signature σ, such that σ will pass the verification algorithm as a valid signature for m with respect to the given public key. This kind of security is usually proven based on the game-based definition as presented in [6].

The second approach is via emulating an "ideal signature process" in an ideal-process-based general framework for analyzing security of protocols [3]. In this approach, one first formulates an "ideal signature functionality" that captures the desired security properties of signature algorithms in an abstract way. A signature algorithm is said to be secure if it "emulates" the ideal signature functionality. Emulation means that for any adversary Adv attacking a real protocol execution, there should exist an "ideal process adversary" or simulator Sim, that causes the outputs of the parties in the ideal process to be essentially the same as the outputs of the parties in a real execution [1]. The reason that why this ideal-model based analysis of signature algorithms is of interest is that it provides very strong secure composability properties. Furthermore, the equivalence of the two approaches has been proved in [4].

In this section, we prove that Protocol constitutes a secure two-party signing protocol based on SM9-DSA. The proof is under standard assumption using simulation-based definition, i.e. the second approach. And we prove security in the presence of malicious adversaries and static corruptions.

1.2.1 Definition of Security

For a digital signature algorithm, the same public/private key pair is used for signing many messages. So in the security proof of our two-party signing protocol, we formalize an ideal functionality [3] instead of a trusted third party [2] to define the security.

We first define an ideal functionality $\mathcal{F}_{SM9-DSA}$ formally in Figure 1.

The SM9-DSA Functionality $\mathcal{F}_{SM9-DSA}$

Functionality $\mathcal{F}_{SM9-DSA}$ works with two parties A_1 and A_2, as follows:

- Upon receiving $Sign(sid, M, ds_{A_1})$ from A_1, and $Sign(sid, M, ds_{A_2})$ from A_2, if sid has not been used before, compute a SM9-DSA signature (h, S) on M with private key $ds_A = ds_{A_1} \cdot ds_{A_2}$ (by computing the element $g = e(P_1, P_{pub-s})$ of group G_T, choosing a random $r \in [1, N-1]$, computing the element $\omega = g^r$ of group G_T, computing the integer $h = H_2(M||\omega, N)$. computing the integer $l = (r-h) \ mod \ N$, and computing the element $S = l \cdot ds_A$ of group G_1), finally send the signature to both A_1 and A_2.

Figure 1: The SM9-DSA Functionality $\mathcal{F}_{SM9-DSA}$

1.2.2 Proof of Security

Theorem 1 *The protocol securely computes $\mathcal{F}_{SM9-DSA}$ in the ($\mathcal{F}_{zk}$, $\mathcal{F}_{com-zk}$)-hybrid model in the presence of a malicious static adversary, assume that the Paillier encryption scheme is indistinguishable under chosen-plaintext attacks, and ECDLP (Elliptic Curve Discrete Logarithm Problem) is hard.*

Proof 1.1 *Below, we first prove the security for the case that A_1 is corrupted. Let Adv be an adversary who has corrupted A_1, we construct a simulator Sim working as follows:*

- *Simulating - corrupted A_1:*

(1) Upon input $Sign(sid, M, ds_{A_1})$, simulator Sim sends $Sign(sid, M, ds_{A_1})$ to $\mathcal{F}_{SM9-DSA}$ and receives back (h, S).

(2) Sim invokes Adv upon input $Sign(sid, M, ds_{A_1})$ and receives $(\textbf{com-prove}, sid||1, g_1, r_1)$ as Adv intends to send to $\mathcal{F}_{com-zk}^{R_{DL}}$.

(3) Sim computes an element $g = e(P_1, P_{pub-s})$ of group G_T, and verifies that $g_1 = g^{r_1}$. If yes, then it do as follows:

(a) computes another element $t = g^h$ of group G_T and an integer $h_1 = H_1(ID_A||hid, N)$.
(b) computes an element $P = h_1 \cdot P_2 + P_{pub-s}$ of group G_2.
(c) computes elements $u = e(S, P)$ and $\omega = u \cdot t$ of group G_T.
(d) computes $g_2' = \omega^{r_1^{-1}}$.

If no, then Sim just chooses a random g_2'.

(4) Sim internally hands $(\textbf{proof}, sid||2, g_2')$ to Adv as if sent by $\mathcal{F}_{zk}^{R_{DL}}$.

(5) Sim receives $(\textbf{decom-proof}, sid||1, g_1)$ as Adv intends to send to $\mathcal{F}_{com-zk}^{R_{DL}}$, $(\textbf{prove}, sid||1, n, (p_1, p_2))$ as Adv intends to send to $\mathcal{F}_{zk}^{R_{PE}}$, and $(\textbf{prove}, sid||1, (c_1, pk, g_1), (r_1, sk))$ as Adv intends to send to $\mathcal{F}_{zk}^{R_{PEDL}}$.

(6) Sim checks that $g_1 = g^{r_1}$, $pk = n = p_1 \cdot p_2$, and $c_1 = Enc_{pk}(r_1)$. Once one of the three equations does not holds, Sim simulates A_2 aborting and sends abort to the trusted party computing $\mathcal{F}_{SM9-DSA}$. Otherwise, it continues.

(7) Sim chooses a random $r_4 \in [1, N-1]$, computes $c_3' = Enc_{pk}(r_4)$ and $c_4' = (c_3')^{-1} \cdot ds_{A_1}^{-1} \cdot S$, and internally hands c_3' and c_4' to Adv.

The differences between the view of Adv in a real execution and in the simulation includes:

(1) the way that g_2 is generated: in a real execution, A_2 chooses a random r_2 and computes $g_2 = g^{r_2}$; while in the simulation, Sim computes $g_2' = \omega^{r_1^{-1}}$. However, since $\omega = g^r$ where r is randomly chosen, the distribution over g^{r_2} and $\omega^{r_1^{-1}}$ are identical.

(2) Sim simulates A_2 aborting if $g_1 \neq g^{r_1}$ or $pk = n \neq p_1 \cdot p_2$ or $c_1 \neq Enc_{pk}(r_1)$, while in the real execution, A_2 aborts when the ZK proofs of R_{DL} or R_{PE} or R_{PEDL} are not accepted. By the soundness of these proofs, this difference is at most negligible.

(3) the way that c_3 and c_4 are generated: in a real execution, A_2 chooses a random $r_3 \in [1, N-1]$, and computes $c_2 = Enc_{pk}(h)$, $c_3 = Enc_{pk}(r_3(r_1 r_2 - h))$ (using the homomorphic property of Paillier encryption), $c_4 = r_3^{-1} \cdot ds_{A_2}$. While in the simulation, Sim chooses a random $r_4 \in [1, N-1]$, computes $c_3' = Enc_{pk}(r_4)$ and $c_4' = (c_3')^{-1} \cdot ds_{A_1}^{-1} \cdot S$. However, due to the Paillier encryption is indistinguishable under chosen-plaintext attacks, the encryption of r_4 and $r_3(r_1 r_2 - h)$ are indistinguishable since both r_4 and $r_3(r_1 r_2 - h)$ are randomly distributed over $[1, N-1]$. In addition, due to the ECDLP is hard, $r_3^{-1} \cdot ds_{A_2}$ and $(c_3')^{-1} \cdot ds_{A_1}^{-1} \cdot S = (c_3')^{-1} \cdot (r_1 r_2 - h) \cdot ds_{A_2}$ are indistinguishable.

Thus, we can gain the conclusion that the joint distribution of Adv's view and A_2's output in the ideal simulation is statistically close to the distribution in the real execution. Specifically, we have that

$$((g_2', c_3', c_4'), (r, S)) \equiv_c ((g_2, c_3, c_4), (r, S)).$$

That is, for any P.P.T. (probabilistic polynomial time) distinguisher D, we have

$$Pr[D((VIEW_{Adv}(ds_{A_1}), OP(ds_{A_1}, ds_{A_2})), (VIEW_{A_1}(ds_{A_1}), OP(ds_{A_1}, ds_{A_2}))) = 1] \leq \frac{1}{poly(N)}$$

where OP denotes OUTPUT. The proof of this simulation case is completed.

Now, we prove the security for the case that A_2 is corrupted. Let Adv be an adversary who has corrupted A_2, we construct a simulator Sim working as follows:

- *Simulating - corrupted A_2:*

(1) Upon input $Sign(sid, M, ds_{A_2})$, simulator Sim sends $Sign(sid, M, ds_{A_2})$ to $\mathcal{F}_{SM9-DSA}$ and receives back (h, S).

*(2) Sim computes $c_1' = Enc_{pk}(\tilde{r_1})$ for a random $\tilde{r_1} \in [1, N-1]$ with pk sent by A_1, and internally hands Adv the message (**proof-receipt**, $sid||1$) as if sent by $\mathcal{F}_{com-zk}^{R_{DL}}$, and the message (pk, c_1') as if sent by A_1.*

*(3) Sim receives (**prove**, $sid||2, g_2, r_2$) as Adv intends to send to $\mathcal{F}_{zk}^{R_{DL}}$.*

(4) Sim computes an element $g = e(P_1, P_{pub-s})$ of group G_T and verifies that $g_2 = g^{r_2}$. If yes, then it do as follows:

(a) computes another element $t = g^h$ of group G_T and an integer $h_1 = H_1(ID_A||hid, N)$.

(b) computes an element $P = h_1 \cdot P_2 + P_{pub-s}$ of group G_2.

(c) computes elements $u = e(S, P)$ and $\omega = u \cdot t$ of group G_T.

(d) computes that $g_1' = \omega^{r_2}$.

If no, it simulates A_1 aborting and halts.

*(5) Sim hands Adv the message (**decom-proof**, $sid||1, g_1'$) as if sent by $\mathcal{F}_{com-zk}^{R_{DL}}$.*

(6) Sim runs the simulator for the ZK proof of relation R_{PEDL} for common statement (c_1', pk, g_1') and with the residual Adv as verifier.

(7) Sim receives c_3 and c_4 from Adv. Then it computes $c_2 = Enc_{pk}(\tilde{r_1} r_2 - h)$, and verifies that $c_2 \cdot ds_{A_2} = c_3 \cdot c_4$. If yes, Sim outputs the signature; otherwise, Sim simulates A_1 aborting.

The differences between the view of Adv in a real execution and in the simulation includes:

(1) the computing of c_1: in a real execution, $c_1 = Enc_{pk}(r_1)$ where $g_1 = g^{r_1}$; while in the simulation, $c_1' = Enc_{pk}(\tilde{r_1})$ for a random $\tilde{r_1}$. However, because Sim uses the same pk as A_1, the indistinguishability of this simulation follows from a straightforward reduction to the indistinguishability of the Paillier encryption scheme, under chosen-plaintext attacks.

(2) the simulation of the ZK proof of relation R_{PEDL}. The indistinguishability is guaranteed by the ZK property of the proof. See details in [6].

(3) the way that g_1 is generated: in a real execution, A_1 chooses a random r_1 and computes $g_1 = g^{r_1}$; while in the simulation, Sim computes $g_1' = \omega^{r_2^{-1}}$. However, since $\omega = g^r$ where r is randomly chosen, the distribution over g^{r_1} and $\omega^{r_2^{-1}}$ are identical.

Thus, we can gain the conclusion that the distribution of Adv's view in the ideal simulation is statistically close to the distribution in the real execution. As for the output of A_1, in the real execution, A_1 checks that c_3 and c_4 are computed correctly by verifying whether the final signature is valid; while in the simulation, Sim, without knowing sk, can not decrypts c_3. However, Sim can check that c_3 and c_4 are computed correctly by computing $c_2 = Enc_{pk}(\tilde{r_1} r_2 - h)$, and verifying that $c_2 \cdot ds_{A_2} = c_3 \cdot c_4$. If $c_2 \cdot ds_{A_2} = c_3 \cdot c_4$ holds, it means that using c_3 and c_4 sent by Adv can get the correct final signature. A little formally, we have that

$$((c_1', g_1'), (r, S)) \equiv_c ((c_1, g_1), (r, S)).$$

That is, for any P.P.T. distinguisher D, we have

$$Pr[D((VIEW_{Adv}(ds_{A_2}), OP(ds_{A_1}, ds_{A_2})), (VIEW_{A_2}(ds_{A_2}), OP(ds_{A_1}, ds_{A_2}))) = 1] \leq \frac{1}{poly(N)}$$

This completes the proof of this simulation case.

References

[1] Canetti, Lindell Yehuda, Ostrovsky Rafail, et al. Universally composable two-party and multi-party secure computation. Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing, 2002. 494-503

[2] R Canetti. Security and Composition of Multiparty Cryptographic Protocols. Journal of Cryptology, 2000, 13: 143–202

[3] R Canetti. Universally composable security: a new paradigm for cryptographic protocols. In: Proceedings 2001 IEEE International Conference on Cluster Computing, Newport Beach, CA, 2001. 136-145

[4] R Canetti. Universally composable signature, certification, and authentication. In: 17th IEEE Computer Security Foundations Workshop, Pacific Grove, CA, USA, 2004. 219-233

[5] Goldwasser Shafi, Micali Silvio, Rivest R L. A digital signature scheme secure against adaptive chosen-message attacks. Siam Journal on Computing, 1988, 17: 281-308

[6] Lindell Yehuda. Fast secure two-party ECDSA signing. In: Advances in Cryptology, Cham, 2017. 613-644