Manuscript

Xun Zhang Bingsheng Zhang
Zhejiang University, CHN
22221024@zju.edu.cn bingsheng@zju.edu.cn

March 12, 2024

1 Building Blocks

1.1 Schnorr signature

The original Schnorr signature scheme:

We propose a batched version of schnorr signature:

1.2 Commitment scheme to $\mathbb{Z}_q$-vectors

Commitment scheme from [DRZ20], use structured pedersen commitment key. The commitment scheme is in Fig. 3.

The above scheme is perfectly hiding and computationally binding under the DLOG assumption [DRZ20].

1.3 $\sum$-Protocol for opening a linear form

We consider the following relation:

$$\begin{aligned}\mathcal{R} = (P \in \mathbb{G}, L \in \mathcal{L}(\mathbb{Z}_q^n), y \in \mathbb{Z}_q; \mathbf{x} \in \mathbb{Z}_q^n, \gamma \in \mathbb{Z}_q) :\\ P = \mathsf{Com}_{\mathsf{ck}}(\mathbf{x}; \gamma) \ \ \wedge \ \ L(\mathbf{x}) = y\end{aligned}$$

The protocol is in Fig. 4.

1.4 Opening a Committed Linear Form

In above protocol, the communication complexity as well as the verifier complexity is linear due to the last message sent by the prover and the last check performed by the verifier. To improve both complexities, we replace the message sent in the last step with a proof of knowledge.

Defined a new relation $\mathcal{R}_{CLF}$ to capture that, where the new linear form L is defined as $L(\mathbf{z}, \phi) := L(\mathbf{z})$ and the message the prover sends is exactly the witness in this relation:

The Schnorr signature protocol

Common parameters: security parameter λ, a hash function $H : \{0,1\}^* \to \mathbb{Z}_q$, and a message $M \in \{0,1\}^*$.
Protocol:

- $\mathsf{Gen}_{\mathsf{gp}}(1^\lambda)$: take input as security parameter λ, and output a group parameter param.
- The Signer S pick $x \leftarrow \mathbb{Z}_q^*$ and set $y := g^x$, and output $(\mathsf{pk} := (g, y), \mathsf{sk} := (\mathsf{pk}, x))$.
- To sign a message M, signer S do:
  - choose a random k from the allowed set.
  - output random nonce $r := g^k$.
  - calculate the challenge $e := H(r||M)$ and the signature $s := k - xe$. Then output the signature pair $sig := (s, e)$.
- To verify a signature, verifier V do:
  - calculate the prefix randomness $z := g^s y^e$.
  - calculate the signature $e' := H(z||M)$.
  - check if $e = e'$. If it holds, output 1, otherwise output 0.

Figure 1: The Schnorr signature protocol

The batched Schnorr signature protocol

Common parameters: security parameter λ, a hash function $H : \{0,1\}^* \to \mathbb{Z}_q$, the number of the signer n. The signer set $\mathbf{S}$, $|\mathbf{S}| = n$. and a message set $\mathbf{M}$, $|\mathbf{M}| = n$,$M_i \in \{0,1\}^*$.
Protocol:

- $\mathsf{Gen}_{\mathsf{gp}}(1^\lambda)$: take input as security parameter λ, and output a group parameter param.
- The Signer set $\mathbf{S}$, for $i = 1, \ldots, n$, every signer S_i pick $x_i \leftarrow \mathbb{Z}_q^*$ and set $y_i := g^{x_i}$, and output $(\mathsf{pk} := (g, y), \mathsf{sk} := (\mathsf{pk}, x_i))$
- For $i = 1, \ldots, n$, every signer S_i do:
  - choose a random k_i from the allowed set.
  - output random nonce $r_i := g^{k_i}$.
  - calculate the challenge $e_i := H(r_i || M_i)$ and the signature $s_i := k_i - x_i e_i$. Then output the signature pair $sig := (s_i, e_i)$.
- To verify a signature, verifier V do:
  - For $i = 1, \ldots, n$, calculate the prefix randomness $z_i := g^{s_i} y_i^{e_i}$.
  - For $i = 1, \ldots, n$, calculate the signature $e_i' := H(z_i || M_i)$.
  - For $i = 1, \ldots, n$, check if $e_i = e_i'$. If it holds, output 1, otherwise output 0.

Figure 2: The batched Schnorr signature protocol

Commitment scheme

Common parameters: $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h)$ be a bilinear group. length of vector n.

- $\mathsf{Gen}_{\mathsf{gp}}(1^\lambda)$: take input as security parameter λ, and outputs a global parameter param.
- $\mathsf{Gen}_{\mathsf{ped}}(\mathsf{param})$: first samples a l-length vector $\dot{\mathbf{a}} := (\dot{a}_1, \ldots, \dot{a}_l) \leftarrow \mathbb{Z}_q^l$ where $l = \log(n+1)$. Let $\mathbf{a} := (a_1, \ldots, a_n) \leftarrow \mathbb{Z}_q^{n+1}$ be defined as $a_j = \prod_{i=1}^{l} \dot{a}_i^{b_{ji}}$, where $(b_{j1}, \ldots, b_{jl})$ is the binary representation of j.

  Outputs commitment key $ck := (g^{a_1}, \ldots, g^{a_n})$ and verification key $vk := (H^{\dot{a}_1}, \ldots, H^{\dot{a}_l})$.
- $\mathsf{Com}_{\mathsf{ck}}(\mathbf{x}; \gamma)$: outputs a commitment $c := g^{<\mathbf{a},(\mathbf{x}||y)>} = \prod_{i=1}^{n+1}(g^{a_i})^{x_i||y}$
- $\mathsf{Open}_{\mathsf{ck}}(c)$: outputs $x \in \mathbb{Z}_q^n$, $\gamma \in \mathbb{Z}_q$ such that $c := g^{<\mathbf{a},(\mathbf{x}||y)>} = \prod_{i=1}^{n+1}(g^{a_i})^{x_i||y}$.

Figure 3: Commitment scheme

$\sum$-Protocol $\prod_0$ for opening a linear form

Parameters:
Common parameters: $P \in \mathbb{G}, L \in \mathcal{L}(\mathbb{Z}_q^n), y \in \mathbb{Z}_q, P = \mathsf{Com}_{[\mathsf{a}]}(\mathbf{x}; \gamma), L(\mathbf{x}) = y$
P's inputs: $\mathbf{x} \in \mathbb{Z}_q^n, \gamma \in \mathbb{Z}_q$.
Protocol:
1. P samples $r \leftarrow_R \mathbb{Z}_q^n$ and $\rho \leftarrow_R \mathbb{Z}_q$, computes $A = \mathsf{Com}_{\mathsf{ck}}(\mathbf{r}; \rho)$, $t = L(\mathbf{r})$ and sends A, t to V.
2. V samples $c \leftarrow_R \mathbb{Z}_q$ and sends to P.
3. P computes $\mathbf{z} = c\mathbf{x} + \mathbf{r}$ and $\phi = c\gamma + \rho$ and sends $\mathbf{z}, \phi$ to V.
4. V checks if $\mathsf{Com}_{\mathsf{ck}}(\mathbf{z}; \phi) = AP^c$ and $L(\mathbf{z}) = cy + t$, outputs 1 if it holds, outputs 0 otherwise.

Figure 4: $\sum$-Protocol $\prod_0$ for opening a linear form

$$\mathcal{R}_{CLF} = (P \in \mathbb{G}, Q \in \mathbb{G}, y \in \mathbb{Z}_q; \mathbf{x} \in \mathbb{Z}_q^n, L \in \mathcal{L}(\mathbb{Z}_q^n)) : \\ P = \mathsf{Com}_{\mathsf{ck}}(\mathbf{x}) \wedge Q = \mathsf{Com}_{[\mathbf{a}]}(L) \wedge L(\mathbf{x}) = y$$

[DGJ24] improved the protocol for opening a committed linear form, switch the above relation to the reverse linear form relation:

$$\mathcal{R}_{CLF_rev} = (P \in \mathbb{G}, Q \in \mathbb{G}, y \in \mathbb{Z}_q; \mathbf{x} \in \mathbb{Z}_q^n, L \in \mathcal{L}(\mathbb{Z}_q^n)) : \\ P = \mathsf{Com}_{\mathsf{ck}}(\mathbf{x}) \wedge Q = \mathsf{Com}_{\mathsf{ck}}(rev(L)) \wedge L(\mathbf{x}) = y$$

where $rev(L)$ is the reverse of the original L. The new relation corresponds to showing opening of a public commitment P and a public value y, obtained by operating a linear form L on a secret $\mathbb{Z}_q^n$ vector $\mathbf{x}$, where we also have a commitment to the reverse of linear form L which is represented as a vector.

The whole protocol is in Fig. 5.

To remove the message w sent by P, we need to prove the following relation:

$$\mathcal{R} = (P \in \mathbb{G}, y \in \mathbb{Z}_q, L \in \mathcal{L}(\mathbb{Z}_q^n); \mathbf{x} \in \mathbb{Z}_q^n) : \\ P = \mathsf{Com}_{\mathsf{ck}}(\mathbf{x}) \ \wedge \ L(\mathbf{x}) = y$$

and the check computed by the verifier in step 10 corresponds to ensuring that $(R, \mathbf{c}^{n-1}, z; \mathbf{w}) \in \mathcal{R}$.

The following table $\mathbf{L}$ represents the left half of a vector, while $\mathbf{R}$ represents the right half.

So here is another protocol to prove the relation $\mathcal{R}$ in Fig. 6.

2 Basic Idea discussion

- We noticed that in batched Schnorr signature scheme, there exists a linear combination if all signers shares the same challenge.

  For examples, every signer gets the challenge $e := H(r_1||r_2||\ldots||r_n||M_1||M_2||\ldots||M_n)$. And they can compute the signature vector $\mathbf{s} = \mathbf{k} - e\mathbf{x}$. And this linear relation between three vectors can be embedded into the step3 in $\prod_0$. It seems like we can use $\prod_0$ and its succinct version($\prod_1$ with $\prod_2$) to prove that.

  However, this approach requires prover to know every signer's private key x_i, so it is not practical.

- So we propose a batched Schnorr signature scheme with individual challenge e_i for every signer. And the core part of the verification is to check if $z_i = g^{s_i} y_i^{e_i}$ holds, for the challenge e_i can be calculated first.

  To commit the group element, a commitment scheme uses pairing may be helpful[AFGHO10].

Protocol $\prod_1$ for Opening Committed Linear Form (reverse)

Parameters:
Common parameters: $P \in \mathbb{G}, Q \in \mathbb{G}, y \in \mathbb{Z}_q; H^{\dot{\mathbf{a}}} \in \mathbb{G}^l$

- $P = \mathsf{Com}_{\mathsf{ck}}(\mathbf{x}), Q = \mathsf{Com}_{\mathsf{ck}}(rev(L)), L(\mathbf{x}) = y,$
- $n = 2^l, \dot{\mathbf{a}} = (\dot{a}_1, \ldots, \dot{a}_l), \mathbf{a} = (\prod_{i=1}^{l} \dot{a}_i{}^{b_i})_{b_i \in \{0,1\}}$

P's inputs: $(ck \in \mathbb{G}^n, \mathbf{x} \in \mathbb{Z}_q^n, L \in \mathcal{L}(\mathbb{Z}_q^n))$
Protocol:
1. Define $\mathbf{B} \in \mathbb{Z}_q^n$ as $B = rev(L)$, Let $\mathbf{x}(U)$ be a polynomial of degree $(n-1)$ defined with coefficient vector $\mathbf{x} = (x_1, \ldots, x_n)$, such that $\mathbf{x}(U) = \sum_{i=0}^{n-1} x_{i+1} U^i$. Similarly, we define the polynomial $\mathbf{B}(U)$ of degree $(n-1)$ for the vector $\mathbf{B}$.
2. P defines a $(2n-2)$ degree polynomial p by:

$$\mathbf{p}(U) = \mathbf{x}(U) \cdot \mathbf{B}(U) = \sum_{i,j} x_{i+1} B_{j+1} U^{i+j}$$

then P parses the computed polynomial as:

$$\mathbf{p}(U) = \mathbf{p}_L(U) \cdot U^{-1} + y \cdot U^{n-1} + \mathbf{p}_R(U) \cdot U^n$$

where $\mathbf{p}_L$ is a polynomial of degree $(n-1)$ and $\mathbf{p}_R$ is a polynomial of degree $(n-2)$.
3. P computes $A_1 = \mathsf{Com}_{\mathsf{ck}}(\mathbf{p}_L)$ and $A_2 = \mathsf{Com}_{\mathsf{ck}}(\mathbf{p}_R)$ and sends to V.
4. V samples $c \leftarrow_R \mathbb{Z}_q$ and sends to P.
5. P computes the evaluations of the polynomials on the random challenge c as follows, and then sends $z1, z2, z3$ and $z4$ to V: $z1 = \mathbf{x}(c), z2 = \mathbf{B}(c), z3 = \mathbf{p}_L(c), z4 = \mathbf{p}_R(c)$
6. V checks if following holds:

$$z_3 \cdot c^{-1} + y \cdot c^{n-1} + z_4 \cdot c^n = z_1 \cdot z_2$$

7. V samples $t \leftarrow_R \mathbb{Z}_q$ ans sends to P.
8. P sets $w = x + t \cdot B + t^2 \cdot \mathbf{p}_L + t^3 \cdot \mathbf{p}_R$ and sends w to V.
9. P and V both compute the following :

$$R = P \cdot Q^t \cdot A_1^{t^2} \cdot A_2^{t^3}, z = z_1 + t \cdot z_2 + t^2 \cdot z_3 + t^3 \cdot z_4$$

10. V outputs 1 if for $\mathbf{c}^{n-1} = (1, \ldots, c^{n-1})$ the following relation holds, and outputs 0 otherwise:

$$\mathsf{Com}_{\mathsf{ck}}(\mathbf{w}) = R \ \wedge \ < \mathbf{w}, \mathbf{c}^{n-1} >= z$$

Figure 5: Protocol $\prod_1$ for Opening Committed Linear Form (reverse)

Protocol $\prod_2$ for $(R, \mathbf{c}^{n-1}, z; \mathbf{w}) \in \mathcal{R}$

Parameters:

Common parameters: $P \in \mathbb{G}, Q \in \mathbb{G}, y \in \mathbb{Z}_q; H^{\dot{\mathbf{a}}} \in \mathbb{G}^l$

- $R = \mathsf{Com}_{\mathsf{ck}}(\mathbf{w}), L_c = \mathbf{c}^{n-1} = (1, \ldots, c^{n-1}), z = L_c(\mathbf{w}),$
- $n = 2^l, \dot{\mathbf{a}} = (\dot{a}_1, \ldots, \dot{a}_l), \mathbf{a} = (\prod_{i=1}^{l} \dot{a_i}^{b_i})_{b_i \in \{0,1\}}$

P's inputs: $(ck \in \mathbb{G}^n, \mathbf{w} \in \mathbb{Z}_q^n, L_c = \mathbf{c}^{n-1} \in \mathbb{Z}_q^n)$

Protocol:

1. P computes $A_1 = \mathsf{Com}_{\mathsf{ck}_L}(\mathbf{w}_L)$, $A_2 = \mathsf{Com}_{\mathsf{ck}_L}(\mathbf{w}_R)$ and $z' =< \mathbf{w}_L, (L_c)_L >=< \mathbf{w}_L, \mathbf{c}^{(n-1)/2} >$ sends to V.

2. V checks if

$$e\left(\frac{R}{A_1}, g\right) = e\left(A_2, g^{\dot{a}_l}\right)$$

3. V samples $s \leftarrow_R \mathbb{Z}_q$ ans sends to P.

4. P sets $\mathbf{w}' = \mathbf{w}_L + s \cdot \mathbf{w}_R$, $L'_c = s(L_c)_L + (L_c)_R = (s + c^{n/2})c^{n/2-1}$ and implicitly sets $\dot{\mathbf{a}}' = (\dot{a_1}, \ldots, \dot{a_{l-1}})$ and $\mathbf{a}' = \mathbf{a}_L$. So the new commitment key $ck' := g^{\mathbf{a}_L}$.

5. P and V both compute the following :

$$R' = A_1 A_2^s, \;\; d = c^{n/2} \cdot z' + s \cdot z + s^2 \cdot c^{-n/2} \cdot (z - z')$$

6. If $\mathbf{w}' \notin \mathbb{Z}_q^2$, repeat the step1 to step5 with new parameter $\mathsf{Com}_{\mathsf{ck}'}(\mathbf{w'}) = P'$ and $< L'_c, \mathbf{x}' >= d$

7. If $\mathbf{w}' \in \mathbb{Z}_q^2$:

- P sends $\mathbf{w}', L'_c$ to V.
- V outputs 1 if the following holds, and outputs 0 otherwise:

$$\mathsf{Com}_{\mathsf{ck}'}(\mathbf{w'}) = P' \;\wedge\; < L'_c, \mathbf{x}' >= d$$

Figure 6: Protocol $\prod_2$ for $(R, \mathbf{c}^{n-1}, z; \mathbf{w}) \in \mathcal{R}$

The pairing is defined as: $e : \mathbb{G} \times \hat{\mathbb{G}} \rightarrow \mathcal{T}$ To commit the a vector of group element $(c_1, \ldots, c_m)$, the commitment scheme specifies non-trivial group elements $(v, u_1, \ldots, u_m) \in \hat{\mathbb{G}}$ and a random $t \in \mathbb{G}$. The commitment $C = e(t, v) \prod_{j=1}^{m} e(c_j, u_j)$.

$\mathsf{Com}_{v,u}(\mathbf{z}) = \mathsf{Com}_{v,u}(g^{\mathbf{s}} \mathbf{y}^{\mathbf{e}})$?