Cryptography of Hyperledger Indy

Kyle Huang

November 27, 2019

1 Syntax of Hyperledger Indy

The first four steps are similar to the register operation, and the last four steps look like login.[1]

1. Issuer determines a credential schema $\mathcal{S}$: the type of cryptographic signatures used to sign the credentials, the number l of attributes in a credential, the indices $\mathcal{A}_h \subset [1, l] = \{1, 2, ..., l\}$ of hidden attributes, the public key P_k, the non-revocation credential attribute number l_r and non-revocation public key P_r. Then he publishes it on the ledger and announces the attribute semantics.

2. Holder retrieves the credential schema from the ledger and sets the hidden attributes.

3. Holder requests a credential from issuer. He sends hidden attributes in a blinded form to issuer and agrees on the values of known attributes $\mathcal{A}_k \leftarrow [1, l] \backslash \mathcal{A}_h$.

4. Issuer returns a credential pair (C_p, C_{NR}) to holder. The first credential contains the requested l attributes. The second credential asserts the non-revocation status of the first one. Issuer publishes the non-revoked status of the credential on the ledger.

5. Holder approaches verifier. Verifier sends the Proof Request $\mathcal{E}$ to holder. The Proof Request contains the credential schema $\mathcal{S}_E$ and disclosure predicates $\mathcal{D}$. The predicates for attribute m and value V can be of form $m = V$, $m < V$, or $m > V$. Some attributes may be asserted to be the same: $m_i = m_j$.

6. Holder checks that the credential pair he holds satisfies the schema $\mathcal{S}_E$. He retrieves the non-revocation witness from the ledger.

7. Holder creates a proof $\mathcal{P}$ that he has a non-revoked credential satisfying the proof request $\mathcal{E}$ and sends it to verifier.

8. Verifier verifies the proof.

[1] All content refers to Hyperledger Indy HIPE.

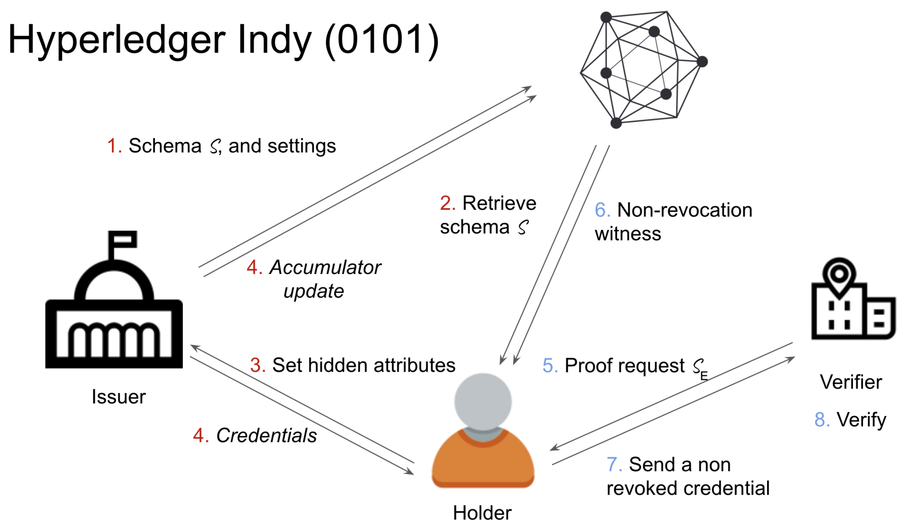

Figure 1: Syntax of Hyperledger Indy

Symbol	Definition
$\mathcal{S}$	Schema, the empty data form with only fields.
(l, l_r)	**Attributes** number and **non-revocation** credential attribute number.
L	The volume of a non-revocation list.
l_a	Message length for all attributes. In Sovrin, $l_a = 256$.
$(\mathcal{A}_k, \mathcal{A}_h)$	The indices of **known attributes** and **hidden attributes** respectively.
	By default, $\{1, 3\} \subset \mathcal{A}_h$ and $\{2\} \subset \mathcal{A}_k$.
(P_k, P_r)	Public keys of **primary credentials** and **non-revocation credentials** resp.
$\mathcal{P}_1$	Correctness proof of P_k.
$(i, \mathcal{H})$	The **index** and **identifier** of a holder in the issuer's view.
$(\mathcal{V}, acc)$	The **indices** and **accumulator** of the current non-revocation list.
(C_P, C_{NR})	The **primary credential** and the **non-revocation credential**.

Table 1: Symbol table

2 Environment setup

Issuer generates the key pair (P_k, s_k) and a proof $\mathcal{P}_1$ through $setup_{PC}(l)$ (Algorithm 1), as well as the non-revocation key pair (P_r, s_r) through $setup_{NR}()$ (Algorithm 3); then, he keeps (s_k, s_r) secret and publishes $(\mathcal{S}, \mathcal{A}_h, l_r, P_k, P_r, \mathcal{P}_1)$ to the ledger. Everyone can verify the correctness of P_k (via proof $\mathcal{P}_1$) through $verify_{P_k}(l, P_k, \mathcal{P}_1)$ (Algorithm 2).

2.1 Primary Credential (CL-Signature)

Algorithm 1 $setup_{PC}(l)$

$p', q' \leftarrow_R \{0,1\}^{1536}$ ▷ p' and q' are prime; $|p'| = |q'| = 1536$
$p \leftarrow 2p' + 1$; $q \leftarrow 2q' + 1$; $n \leftarrow pq$ ▷ p and q are prime
$t \leftarrow_R \mathbb{Z}_n^*$; $S \leftarrow t^2 \pmod n$
$x_z \leftarrow_R \mathbb{Z}_{p'q'}^*$, $Z \leftarrow S^{x_z} \pmod n$
$\{x_{r_i} \leftarrow_R \mathbb{Z}_{p'q'}^*, R_i \leftarrow S^{x_{r_i}} \pmod n\}_{\forall i \in [1,l]}$
$P_k \leftarrow (n, S, Z, \{R_i\}_{\forall i \in [1,l]})$, $s_k \leftarrow (p, q)$
$\tilde{x}_z \leftarrow_R \mathbb{Z}_{p'q'}^*$, $\tilde{Z} \leftarrow S^{\tilde{x}_z} \pmod n$ ▷ correctness proof from here.
$\{\tilde{x}_{r_i} \leftarrow_R \mathbb{Z}_{p'q'}^*, \tilde{R}_i \leftarrow S^{\tilde{x}_{r_i}} \pmod n\}_{\forall i \in [1,l]}$
$c \leftarrow H_1(Z || \tilde{Z} || \{R_i, \tilde{R}_i\}_{\forall i \in [1,l]})$ ▷ H_1 is by default SHA2-256
$\hat{x}_z \leftarrow \tilde{x}_z + c \cdot x_z$; $\{\hat{x}_{r_i} \leftarrow \tilde{x}_{r_i} + c \cdot x_{r_i}\}_{\forall i \in [1,l]}$
$\mathcal{P}_1 \leftarrow (c, \hat{x}_z, \{\hat{x}_{r_i}\}_{\forall i \in [1,l]})$
return $(P_k, s_k, \mathcal{P}_1)$

Algorithm 2 $verify_{P_k}(l, P_k, \mathcal{P}_1)$

$(n, S, Z, \{R_i\}_{\forall i \in [1,l]}) \leftarrow P_k$; $(c, \hat{x}_z, \{\hat{x}_{r_i}\}_{\forall i \in [1,l]}) \leftarrow \mathcal{P}_1$
$\tilde{Z} \leftarrow Z^{-c} S^{\hat{x}_z}$; $\{\tilde{R}_i \leftarrow R_i^{-c} S^{\hat{x}_{r_i}}\}_{\forall i \in [1,l]} \pmod n$
return $c == H_1(Z || \tilde{Z} || \{R_i, \tilde{R}_i\}_{\forall i \in [1,l]})$

2.2 Non-Revocation Credential

Algorithm 3 $setup_{NR}()$

$\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ ▷ pick a type-III pairing where $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = q$
$g \leftarrow_R \mathbb{G}_1$; $g' \leftarrow_R \mathbb{G}_2$
$h, h_0, h_1, h_2, \tilde{h} \leftarrow_R \mathbb{G}_1$; $u, \hat{h} \leftarrow_R \mathbb{G}_2$
$sk \leftarrow_R \mathbb{Z}_q^*$, $pk \leftarrow g^{sk}$; $x \leftarrow_R \mathbb{Z}_q^*$, $y \leftarrow \hat{h}^x$
$P_r \leftarrow (h, h_0, h_1, h_2, \tilde{h}, \hat{h}, u, pk, y)$, $s_r \leftarrow (sk, x)$
return (P_r, s_r)

2.3 CKS Accumulator

Issuer creates a new accumulator using $setup_{Acc}(L, P_r)$ (Algorithm 4).

Algorithm 4 $setup_{Acc}(L, P_r)$

$r \leftarrow_R \mathbb{Z}_q^*$; $\{g_i \leftarrow g^{r^i}, g'_i \leftarrow g'^{r^i}\}_{\forall i \in [1,2L] \setminus \{L+1\}}$
$z \leftarrow e(g, g')^{r^{L+1}}$; $V \leftarrow \emptyset$; $acc \leftarrow 1$
$P_a \leftarrow (z, \{g_i, g'_i\}_{\forall i \in [1,2L] \setminus \{L+1\}})$, $s_a \leftarrow r$
$\triangleright$ issuer publishes $(P_a, \mathcal{V})$ on the ledger with identifier $ID_a \leftarrow z$.
return $(P_a, s_a, \mathcal{V}, acc)$

3 Credential Issuance

Let $i < L$ and $\mathcal{H}$ be the index and identifier of the holder in the issuer's system, respectively. The holder acquires the schema $\mathcal{S}$, indices $\mathcal{A}_h$ and public keys (P_k, P_r) from the ledger in addition to a random number n_0 and the identifier $\mathcal{H}$ from the issuer; then he sets the hidden attribute $\{m_i\}_{\forall i \in \mathcal{A}_h}$. The credential issuance process is interactive, which follows:

1. The holder computes a temporary result (P_h, s_h) by excuting (Algorithm 5) $issue_1(\mathcal{S}, \mathcal{A}_h, \{m_i\}_{\forall i \in \mathcal{A}_h}, n_0, \mathcal{H}, P_k, P_r)$; then, he keeps s_h private and sends $(P_h, \{m_i\}_{\forall i \in \mathcal{A}_k})$ to the issuer.

2. On receiving $(P_h, \{m_i\}_{\forall i \in \mathcal{A}_k})$ from the holder, the issuer firstly verifies P_h through $verify_{P_h}(P_k, P_h)$ (Algorithm 6). If it passes, the issuer fetches the current non-revoked indices $\mathcal{V}$ and accumulator acc on the ledger, and runs $issue_2(i, \mathcal{H}, \mathcal{V}, acc, \{m_i\}_{\forall i \in \mathcal{A}_k}, P_k, s_k, P_r, s_r, P_a, s_a, P_h)$ (Algorithm 7) to generate $(P_{PC}, P_{NR}, \mathcal{V}, acc)$. Finally, the issuer stores the holder's information and index i in issue's local database; then, he updates acc and $\mathcal{V}$ on the ledger and returns (P_{PC}, P_{NR}) to the holder.

3. While receiving (P_{PC}, P_{NR}) from the issuer, the holder firstly runs Algorithm 8 to verify P_{NR}. If `True` $\leftarrow verify_{P_{NR}}(acc, \mathcal{H}, s_h, P_r, P_{NR})$, the holder excutes Algorithm 9 $issue_3(P_k, P_h, s_h, P_{PC}, P_{NR})$ to do more verifications. If all verification pass, the holder keeps the returned credential (C_P, C_{NR}).

4 Credential Revocation

The revocation process is quite straightforward. The issuer fetches the current non-revoked indices $\mathcal{V}$ and accumulator acc on the ledger. Then, he revokes user with index i via Algorithm 10 and updates $(\mathcal{V}', acc')$ on the ledger after running $(\mathcal{V}', acc') \leftarrow revoke(\mathcal{V}, acc, i)$.

Algorithm 5 $issue_1(\mathcal{S}, \mathcal{A}_h, \{m_i\}_{\forall i \in \mathcal{A}_h}, n_0, \mathcal{H}, P_k, P_r)$

$\{\tilde{m}_i \leftarrow_R \{0,1\}^{593}\}_{\forall i \in \mathcal{A}_h}$ ▷ primary credential
$v' \leftarrow_R \{0,1\}^{3152}$; $\tilde{v}' \leftarrow_R \{0,1\}^{3488}$
$(n, S, Z, \{R_i\}_{\forall i \in [1,l]}) \leftarrow P_k$
$U \leftarrow S^{v'} \prod_{i \in \mathcal{A}_h} R_i^{m_i}$; $\tilde{U} \leftarrow S^{\tilde{v}'} \prod_{i \in \mathcal{A}_h} R_i^{\tilde{m}_i}$
$c = H(U||\tilde{U}||n_0)$; $n_1 \leftarrow_R \{0,1\}^{80}$
$\hat{v} \leftarrow \tilde{v} + c \cdot v$; $\{\hat{m}_i \leftarrow \tilde{m}_i + c \cdot m_i\}_{\forall i \in \mathcal{A}_h}$
$(h, h_0, h_1, h_2, \tilde{h}, \hat{h}, u, pk, y) \leftarrow P_r$ ▷ non-revocation credential
$s' \leftarrow_R \mathbb{Z}_q^*$, $U_r \leftarrow h_2^{s'}$
$P_h \leftarrow (U, c, \hat{v}', \{m_i\}_{\forall i \in \mathcal{A}_h}, n_1, U_r)$, $s_h \leftarrow (v', s')$
return (P_h, s_h)

Algorithm 6 $verify_{P_h}(P_k, P_h)$

$(n, S, Z, \{R_i\}_{\forall i \in [1,l]}) \leftarrow P_k$; $(U, c, \hat{v}', \{m_i\}_{\forall i \in \mathcal{A}_h}, n_1, U_r) \leftarrow P_h$
$\tilde{U} \leftarrow U^{-c} S^{\hat{v}'} \prod_{i \in \mathcal{A}_h} S^{\hat{m}_i} R_i^{-c} \pmod{n}$
return $c == H(U||\tilde{U}||n_0)$

Algorithm 7 $issue_2(i, \mathcal{H}, \mathcal{V}, acc, \{m_i\}_{\forall i \in \mathcal{A}_k}, P_k, s_k, P_r, s_r, P_a, s_a, P_h)$

$m_2 \leftarrow H(i||\mathcal{H})$ ▷ the primary credential
$v'' \leftarrow \{0,1\}^{2723}$; $e'' \leftarrow \{0,1\}^{596}$ ▷ $|v''| = 2723$, $|e| = 596$ and e is prime
$(n, S, Z, \{R_i\}_{\forall i \in [1,l]}) \leftarrow P_k$; $(U, c, \hat{v}', \{m_i\}_{\forall i \in \mathcal{A}_h}, n_1, U_r) \leftarrow P_h$
$Q \leftarrow Z(US^{v''} \prod_{i \in \mathcal{A}_k} R_i^{m_i})^{-1} \pmod{n}$; $r \leftarrow_R \mathbb{Z}_{p'q'}^*$
$A \leftarrow Q^{e^{-1} \pmod{p'q'}}$; $\hat{A} \leftarrow Q^r \pmod{n}$
$c' \leftarrow H(Q||A||\hat{A}||n_1)$; $s_e \leftarrow r - c'e^{-1}$
$P_{PC} \leftarrow (\{m_i\}_{\forall i \in \mathcal{A}_k}, A, e, v'', s_e, c')$
$s'', c \leftarrow_R \mathbb{Z}_q^*$ ▷ non-revocation credential
$(h, h_0, h_1, h_2, \tilde{h}, \hat{h}, u, pk, y) \leftarrow P_r$, $(sk, x) \leftarrow s_r$
$(z, \{g_i, g_i'\}_{\forall i \in [1,2L] \setminus \{L+1\}}) \leftarrow P_a$, $r \leftarrow s_a$
$\sigma \leftarrow (h_0 h_1^{m_2} U_r g_i h_2^{s''})^{(x+c)^{-1}}$; $\sigma_i \leftarrow g'^{(sk+r^i)^{-1}}$; $u_i \leftarrow u^{r^i}$
$w \leftarrow \prod_{j \in \mathcal{V}} g'_{L+1+i-j}$; $\mathcal{V} \leftarrow \mathcal{V} \cup \{i\}$, $acc \leftarrow acc \cdot g'_{L+1-i}$
$wit_i \leftarrow (\sigma_i, u_i, g_i, w, \mathcal{V})$
$P_{NR} \leftarrow (I_A, \sigma, c, s'', wit_i, g_i, g_i', i)$
return $(P_{PC}, P_{NR}, \mathcal{V}, acc)$

Algorithm 8 $verify_{P_{NR}}(acc, \mathcal{H}, s_h, P_r, P_{NR})$

$(h, h_0, h_1, h_2, \tilde{h}, \hat{h}, u, pk, y) \leftarrow P_r$; $(I_A, \sigma, c, s'', wit_i, g_i, g'_i, i) \leftarrow P_{NR}$
$(\sigma_i, u_i, g_i, w, \mathcal{V}) \leftarrow wit_i$; $(v', s') \leftarrow s_h$
$s \leftarrow s' + s''$; $m_2 \leftarrow H(i||\mathcal{H})$
if $e(g_i,\, acc)(e(g,\, w))^{-1} \neq z$ **then**
    return `False`
else if $e(pk \cdot g_i,\, \sigma_i) \neq e(g,\, g')$ **then**
    return `False`
else if $e(\sigma,\, y \cdot \hat{h}^c) \neq e(h_0 h_1^{m_2} h_2^s \cdot g_i,\, \hat{h})$ **then**
    return `False`
else
    return `True`
end if

Algorithm 9 $issue_3(P_k, P_h, s_h, P_{PC}, P_{NR})$

$(n, S, Z, \{R_i\}_{\forall i \in [1,l]}) \leftarrow P_k$; $(U, c, \hat{v}', \{m_i\}_{\forall i \in \mathcal{A}_h}, n_1, U_r) \leftarrow P_h$
$(\{m_i\}_{\forall i \in \mathcal{A}_k}, A, e, v'', s_e, c') \leftarrow P_{PC}$; $(I_A, \sigma, c, s'', wit_i, g_i, g'_i, i) \leftarrow P_{NR}$
if e is not prime OR $e \notin [2^{596}, 2^{596} + 2^{119}]$ **then**
    return null
end if
$(v', s') \leftarrow s_h$; $v \leftarrow v' + v''$; $s \leftarrow s' + s''$
$Q \leftarrow Z(S^v \prod_{i \in (\mathcal{A}_k \cup \mathcal{A}_h)} R_i^{m_i})^{-1} \pmod n$
if $Q \neq A^e$ **then**
    return null
end if
$\hat{A} \leftarrow A^{c' + s_e \cdot e}$
if $c' \neq H(Q||A||\hat{A}||n_1)$ **then**
    return null
else
    $C_P \leftarrow (\{m_i\}_{\forall i \in (\mathcal{A}_k \cup \mathcal{A}_h)}, A, e, v)$; $C_{NR} \leftarrow (I_A, \sigma, c, s, wit_i, g_i, g'_i, i)$
    return (C_P, C_{NR})
end if

Algorithm 10 $revoke(\mathcal{V}, acc, i)$

$\mathcal{V} \leftarrow \mathcal{V} \backslash \{i\}$
$acc \leftarrow acc \cdot (g'_{L+1-i})^{-1}$
return $(\mathcal{V}, acc)$