- Proxy signature verification: The verifier verifies whether

$$\hat{e}(U + H_2(m,U)(h(\omega)P + y_o), \sigma_p) = \hat{e}(y_p, y_p).$$

- Security: Security of the scheme is based on CDHP in the random oracle model.

5.4.5 Lu, Cao and Dong (2006)

Proposed a designated verifier proxy signature scheme.

Assumption: CDHP is hard.

- Alice computes her public key $y_o = H(ID_o)$, where ID_o is her identity. Then, Alice obtains her private key $x_o \leftarrow \mathcal{KG}_{cdhp}(\textbf{params-cdhp}, y_o)$ from KGC.
- Bob computes his public key $y_p = H(ID_p)$, where ID_p is his identity. Then, Bob obtains his private key $x_p \leftarrow \mathcal{KG}_{cdhp}(\textbf{params-cdhp}, y_p)$ from KGC.
- Martin (a designated verifier) computes his public key $y_m = H(ID_m)$, where ID_m is his identity. Then, Martin obtains his private key $x_m \leftarrow \mathcal{KG}_{cdhp}(\textbf{params-cdhp}, y_m)$ from KGC.
- Delegation capability generation: Alice generates delegation capability σ_o as $\sigma_o = x_o H(\omega)$.
- Delegation capability verification: Bob accepts σ_o if and only if $\hat{e}(\sigma_o, P) = \hat{e}(H(\omega), y_o)$.
- Proxy key generation: Bob computes $\rho_p = \sigma_o + x_p H(\omega)$.
- Proxy signature generation: To generate a designated verifier proxy signature for Martin on message m, Bob does the following:
  - Picks $k_p \in \mathbb{Z}^*_{q-1}$ and computes $r_p = k_p y_m$.
  - Computes $\sigma_p = k_p(y_o + y_p) - H_2(m, r_p) \cdot \rho_p$, where $H_2 : \{0,1\}^* \times G_1 \rightarrow \mathbb{Z}^*_q$.
  - The proxy signature of message m is the tuple $(r_p, \sigma_p, (\omega, m))$.
- Proxy signature verification: The verifier accepts the proxy signature of message m if and only if $\hat{e}(y_o + y_p, r'_p) = \hat{e}(\sigma_p, P) \cdot \hat{e}(H(\omega), y_o + y_p)^{H_2(m,r_p)}$, where $r'_p = \frac{1}{x_m} \cdot r_p$.
- Security: Security of the scheme is based on the CDHP in the random oracle model. But, the scheme requires secure channel for proxy delivery.

5.4.6 Das, Saxena and Phatak (2007)

Proposed a proxy signature scheme based on Hess signature scheme that provides effective proxy revocation mechanism and avoids key escrow problem.

Assumption: CDHP is hard.

- Alice computes her public key $y_o = H(ID_o)$, where ID_o is her identity. Then, Alice generates her private key $x_o \leftarrow \mathcal{KG}_{cdhp}(\textbf{params-cdhp}, y_o)$.
- Bob computes his public key $y_p = H(ID_p)$, where ID_p is his identity. Then, Bob generates his private key $x_p \leftarrow \mathcal{KG}_{cdhp}(\textbf{params-cdhp}, y_p)$.
- Delegation capability generation: Alice computes $\sigma_o = (s_o + b_o H'(\omega, y_o, y_p))$, and $\psi_o = b_o P$. Here, b_o is secret to Alice only and $H' : \{0,1\}^* \times G_1 \times G_1 \rightarrow G_1$.
- Delegation capability verification: Bob accepts σ_o if and only if

$$\hat{e}(s_o, P) = \hat{e}(\psi_o, H'(\omega, y_o, y_p)) \cdot \hat{e}(y_o, Reg_o),$$

  where $Reg_o = sb_o P$, registration token published by the KGC.
- Proxy key generation: Bob computes $\rho_p = s_o + s_p + b_p H'(\omega, y_o, y_p)$. Here, b_p is secret to the proxy signer only.
- Proxy signature generation: To sign a message m, Bob does the following.
  - Selects a random $r \in \mathbb{Z}^*_q$ and compute $R = rP$.
  - Computes $a = h(m, R, y_p)$ and $\psi_p = b_p P$, where $h : \{0,1\}^* \times G_1 \times G_1 \rightarrow \{0,1\}^*$.
  - Computes $\sigma_p = (r + a)^{-1} \rho_p$. The proxy signature of message m is $(\omega, m, R, \sigma_p, \psi_o, \psi_p, y_o, y_p)$.
- Proxy signature verification: The proxy signature is valid if and only if

$$\begin{aligned} & \hat{e}(R + h(m, R, y_p)P, \sigma_p) \\ = \; & \hat{e}(\psi_o + \psi_p, H'(\omega, y_o, y_p)) \cdot \hat{e}(y_o, Reg_o) \cdot \hat{e}(y_p, Reg_p). \end{aligned}$$

- Security: The scheme is secure and does not require secure channel in key issuance stage.

6 Concluding Remarks

We have reviewed a few seminal works on proxy signatures with respect to different security assumptions. In order to give a concise picture of the schemes highlighting the important features and security aspects at a glance, we compare them in the following tables. The Table 4 depicts the DLP-based schemes, Table 5 depicts the RSA-based schemes, and Table 6 depicts the Pairing-based schemes. We note that the computational complexity of the schemes in a same table more or less similar, as their underlying security is based on the same cryptographic primitive. It is observed that many times, a paper typically breaks a previous scheme and proposes a new one, which someone breaks later and, in turn, proposes a new one, and so on. Most of such work, though quite important and useful, essentially provides an incremental advance to the same basic theme. Consequently, we believe