Let $F_{p^2}$ be a field $F_p[i]/(i^2+1)$. We define a set $C_2$ with

$$C_2 \equiv \{(X,Y) \in F_{p^2} \times F_{p^2} \mid Y^2 = X^3 + 3(i+9)^{-1}\} \cup \{(0,0)\} \tag{251}$$

We define a binary operation + and scalar multiplication · with the same equations (248), (249) and (250). $(C_2,+)$ is also known to be a group. We define $P_2$ in $C_2$ with

$$\begin{aligned} P_2 \equiv\ & (11559732032986387107991004021392285783925812861821192530917403151452391805634 \times i \\ & +10857046999023057135944570762232829481370756359578518086990519993285655852781, \\ & 4082367875863433681332203403145435568316851327593401208105741076214120093531 \times i \\ & +8495653923123431417604973247489272438418190587263600148770280649306958101930) \end{aligned} \tag{252}$$

We define $G_2$ to be the subgroup of $(C_2,+)$ generated by $P_2$. $G_2$ is known to be the only cyclic group of order $q$ on $C_2$. For a point $P$ in $G_2$, we define $\log_{P_2}(P)$ be the smallest natural number $n$ satisfying $n \cdot P_2 = P$. With this definition, $\log_{P_2}(P)$ is at most $q-1$.

Let $G_T$ be the multiplicative abelian group underlying $F_{q^{12}}$. It is known that a non-degenerate bilinear map $e : G_1 \times G_2 \to G_T$ exists. This bilinear map is a type three pairing. There are several such bilinear maps, it does not matter which is chosen to be $e$. Let $P_T = e(P_1, P_2)$, $a$ be a set of $k$ points in $G_1$, and $b$ be a set of $k$ points in $G_2$. It follows from the definition of a pairing that the following are equivalent

$$\log_{P_1}(a_1) \times \log_{P_2}(b_1) + \cdots + \log_{P_1}(a_k) \times \log_{P_2}(b_k) \equiv 1 \mod q \tag{253}$$

$$\prod_{i=0}^{k} e(a_i, b_i) = P_T \tag{254}$$

Thus the pairing operation provides a method to verify (253).

A 32 byte number $\mathbf{x} \in \mathbf{P}_{256}$ might and might not represent an element of $F_{\mathrm{p}}$.

$$\delta_{\mathrm{p}}(\mathbf{x}) \equiv \begin{cases} \mathbf{x} & \text{if } \mathbf{x} < p \\ \varnothing & \text{otherwise} \end{cases} \tag{255}$$

A 64 byte data $\mathbf{x} \in \mathbf{B}_{512}$ might and might not represent an element of $G_1$.

$$\delta_1(\mathbf{x}) \equiv \begin{cases} g_1 & \text{if } g_1 \in G_1 \\ \varnothing & \text{otherwise} \end{cases} \tag{256}$$

$$g_1 \equiv \begin{cases} (x,y) & \text{if } x \neq \varnothing \wedge y \neq \varnothing \\ \varnothing & \text{otherwise} \end{cases} \tag{257}$$

$$x \equiv \delta_{\mathrm{p}}(\mathbf{x}[0..31]) \tag{258}$$

$$y \equiv \delta_{\mathrm{p}}(\mathbf{x}[32..63]) \tag{259}$$

A 128 byte data $\mathbf{x} \in \mathbf{B}_{1024}$ might and might not represent an element of $G_2$.

$$\delta_2(\mathbf{x}) \equiv \begin{cases} g_2 & \text{if } g_2 \in G_2 \\ \varnothing & \text{otherwise} \end{cases} \tag{260}$$

$$g_2 \equiv \begin{cases} ((x_0 i + y_0), (x_1 i + y_1)) & \text{if } x_0 \neq \varnothing \wedge y_0 \neq \varnothing \wedge x_1 \neq \varnothing \wedge y_1 \neq \varnothing \\ \varnothing & \text{otherwise} \end{cases} \tag{261}$$

$$x_0 \equiv \delta_{\mathrm{p}}(\mathbf{x}[0..31]) \tag{262}$$

$$y_0 \equiv \delta_{\mathrm{p}}(\mathbf{x}[32..63]) \tag{263}$$

$$x_1 \equiv \delta_{\mathrm{p}}(\mathbf{x}[64..95]) \tag{264}$$

$$y_1 \equiv \delta_{\mathrm{p}}(\mathbf{x}[96..127]) \tag{265}$$

We define $\Xi_{\mathtt{SNARKV}}$ as a precompiled contract which checks if (253) holds, for intended use in zkSNARK verification.