

```
graph TD
    subgraph EchoPulse Key Derivation Path (SK + r → K)
        direction LR
        v0((v0)) -->|s1| v1
        v1 -->|s2| v2
        v2 -->|s3| v3
        v3 -->|s4| v4
        v4 -->|s5| v5
        v5 -->|s6| v6
        v6 -->|s<0xE2><0x82><0x87>| v7
        v7 -->|s<0xE2><0x82><0x88>| v8
        v8 -->|...| v_intermediate_sk
        v_intermediate_sk -->|s26| v_prev_priv
        v_prev_priv -->|s2<0xE2><0x82><0x87>| v_priv
        v_priv((v2<0xE2><0x82><0x87> = v_priv))
    end
    style v_priv fill:#ccf,stroke:#333,stroke-width:2px
    subgraph Public Payload (r)
        direction LR
        v_priv -->|r1| r1
        r1 -->|r2| r2
        r2 -->|r3| r3
        r3 -->|r4| r4
        r4 -->|r5| r5
        r5 -->|r6| r6
        r6 -->|r<0xE2><0x82><0x87>| r7
        r7 -->|r<0xE2><0x82><0x88>| r8
        r8 -->|...| r_intermediate
        r_intermediate -->|r26| r_prev_enc
        r_prev_enc -->|r2<0xE2><0x82><0x87>| v_enc
        v_enc((v2<0xE2><0x82><0x87> = v_enc))
    end
    style v_enc fill:#f9f,stroke:#333,stroke-width:2px
    subgraph SHA3-256 Hashing
        direction LR
        input_hash[v_enc || r] --> hash_output(K)
    end
    v_enc --> input_hash
    r_payload[r] --o input_hash
    end
    title EchoPulse Symbolic Key Path Graph (SK + r → K)
```

Code-Snippet

% This is a placeholder for the PDF output of the Mermaid diagram. % To generate the PDF, you would typically use a tool that can render % Mermaid syntax, such as: % - The Mermaid Live Editor (<https://mermaid.live-editor.com/>) % - A command-line tool like mmdc (Mermaid CLI) % - Integration within a document processing system that supports Mermaid. % Using the Mermaid Live Editor, you would paste the code block above % into the editor and then export the diagram as a PDF or SVG file. % For example, if using mmdc: % 1. Save the Mermaid code to a file named "echo_path.mmd". % 2. Run the command: "mmdc -i echo_path.mmd -o echo_path.pdf" % (assuming mmdc is installed and in your system's PATH). % The resulting "echo_path.pdf" file would contain the visual diagram.

```
\documentclass{article}
\usepackage{graphicx}
\usepackage{svg}
\begin{document}
\section*{EchoPulse Symbolic Key Path Graph (SK + r → K)}
\includegraphics[width=\textwidth]{echo_path.svg}
\end{document}
```

Explanation of the Mermaid Diagram:

graph TD: Defines a top-to-bottom directed graph.

subgraph EchoPulse Key Derivation Path (SK + r → K): Creates a visual grouping for the entire process. direction LR sets the internal layout to left-to-right.

v0((v0)): Defines a node representing the initial state v_0, with double parentheses indicating it's a starting point.

-->|s1| v1: Represents a directed edge (transition) from state v0 to v1, labeled with the symbol s_1.

v1 -->|s2| v2 through v8 -->|...| v_intermediate_sk: Illustrates the sequence of transitions driven by the symbols of the secret key SK.

v_intermediate_sk -->|s26| v_prev_priv and v_prev_priv -->|s2<0xE2><0x82><0x87>|

v_priv((v2<0xE2><0x82><0x87> = v_priv)): Shows the final transitions leading to the private key state v_priv. Double parentheses and styling highlight this node.

subgraph Public Payload (r): Creates a subgraph for the path driven by the random payload r, also with a left-to-right layout.

v_priv -->|r1| r1 through r_prev_enc -->|r2<0xE2><0x82><0x87>|

v_enc((v2<0xE2><0x82><0x87> = v_enc)): Depicts the transitions from v_priv using the symbols of r to reach the encapsulation state v_enc, which is also highlighted.

subgraph SHA3-256 Hashing: A subgraph representing the hashing operation.

input_hash[v_enc || r]: A rectangular node representing the concatenation of the encoded v_enc and r.

hash_output(K): A rounded node representing the output of the SHA3-256 hash function, the shared key K.

v_enc --> input_hash and r_payload[r] --o input_hash: Arrows indicating the inputs to the hashing process. r_payload is a separate node to explicitly show r as an input.