Function Secret Sharing, Distributed Point Functions

Slides adapted from here, here

Function secret sharing

Function secret sharing

- Allows a dealer to split a function f into function shares f_i such that for any input $x, f(x) = \sum_i^n f_i(x)$

Function secret sharing

- Allows a dealer to split a function f into function shares f_i such that for any input $x, f(x) = \sum_i^n f_i(x)$
- Function shares must be

Function secret sharing

- Allows a dealer to split a function f into <u>function shares</u> f_i such that for any input $x, f(x) = \sum_i^n f_i(x)$

- Function shares must be

  - Succinct - otherwise one could trivially share the truth table of f

Function secret sharing

- Allows a dealer to split a function f into function shares f_i such that for any input $x, f(x) = \sum_{i}^{n} f_i(x)$

- Function shares must be
  - Succinct - otherwise one could trivially share the truth table of f
  - Secret - function shares should not reveal anything about the function f

Function secret sharing

- Allows a dealer to split a function f into <u>function shares</u> f_i such that for any input $x, f(x) = \sum_{i}^{n} f_i(x)$

- Function shares must be
  - Succinct - otherwise one could trivially share the truth table of f
  - Secret - function shares should not reveal anything about the function f

- Setting: multiple servers with some collusion threshold, each holding a copy of the full dataset

Function secret sharing

Function secret sharing

- Gen$(1^\lambda, \hat{f})$ is a PPT key generation algorithm, which outputs an m-tuple of keys $(k_1, \cdots, k_m)$

- Eval(i, k_i, x) is a polynomial-time evaluation algorithm, which on input $i \in [m]$ (party index), k_i (key defining the function share f_i), outputs a group element $y_i \in \mathbb{G}$ (the value of $f_i(x)$)

Distributed point functions (DPFs)

Distributed point functions (DPFs)

- A point function $f_{\alpha,\beta}$ for $\alpha = \in \{0,1\}^n$ and $\beta \in \mathbb{G}$, is defined to be the function $f : \{0,1\}^n \to \mathbb{G}$ such that

Distributed point functions (DPFs)

- A point function $f_{\alpha,\beta}$ for $\alpha = \in \{0,1\}^n$ and $\beta \in \mathbb{G}$, is defined to be the function $f : \{0,1\}^n \to \mathbb{G}$ such that
  - $f(\alpha) = \beta$

Distributed point functions (DPFs)

- A <u>point function</u> $f_{\alpha,\beta}$ for $\alpha = \in \{0,1\}^n$ and $\beta \in \mathbb{G}$, is defined to be the function $f : \{0,1\}^n \rightarrow \mathbb{G}$ such that
  - $f(\alpha) = \beta$
  - $f(x) = 0$ for $x \neq \alpha$

DPF construction

DPF construction

- Let $G : \{0,1\}^\lambda \rightarrow \{0,1\}^{2\lambda}$ be a length-doubling pseudorandom generator

DPF construction

- Let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ be a length-doubling pseudorandom generator
- GGM-style pseudorandom function:

DPF construction

- Let $G : \{0,1\}^\lambda \rightarrow \{0,1\}^{2\lambda}$ be a length-doubling pseudorandom generator
- GGM-style pseudorandom function:
  - Given an output of $G(s)$, can divide into two halves where $G(s) = G_0(s) || G_1(s)$

DPF construction

- Let $G : \{0,1\}^{\lambda} \rightarrow \{0,1\}^{2\lambda}$ be a length-doubling pseudorandom generator
- GGM-style pseudorandom function:
  - Given an output of $G(s)$, can divide into two halves where $G(s) = G_0(s) || G_1(s)$
  - If s is the root, then recursively applying G will result in a tree

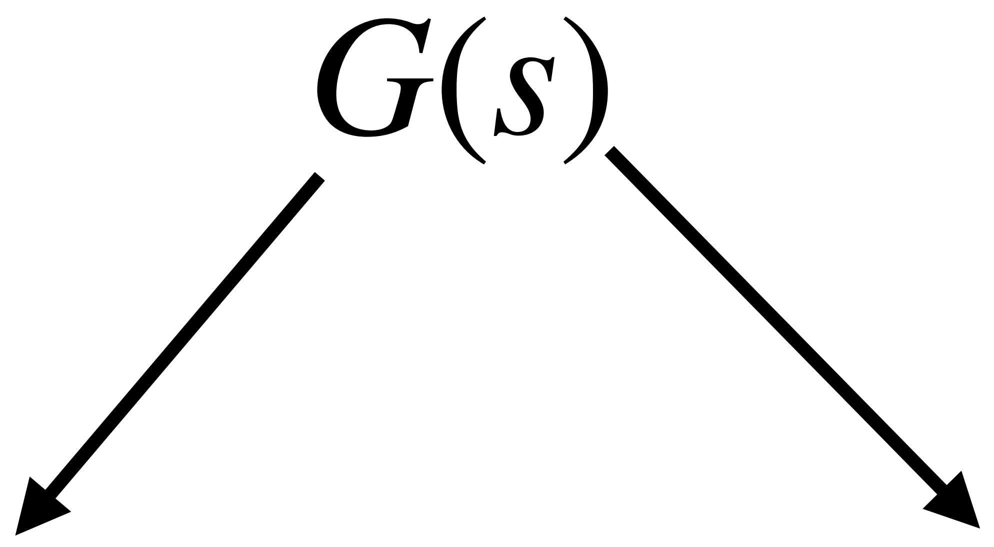
DPF construction

- Let $G : \{0,1\}^{\lambda} \rightarrow \{0,1\}^{2\lambda}$ be a length-doubling pseudorandom generator
- GGM-style pseudorandom function:
  - Given an output of $G(s)$, can divide into two halves where $G(s) = G_0(s)\,||\,G_1(s)$
  - If s is the root, then recursively applying G will result in a tree

$G(s)$

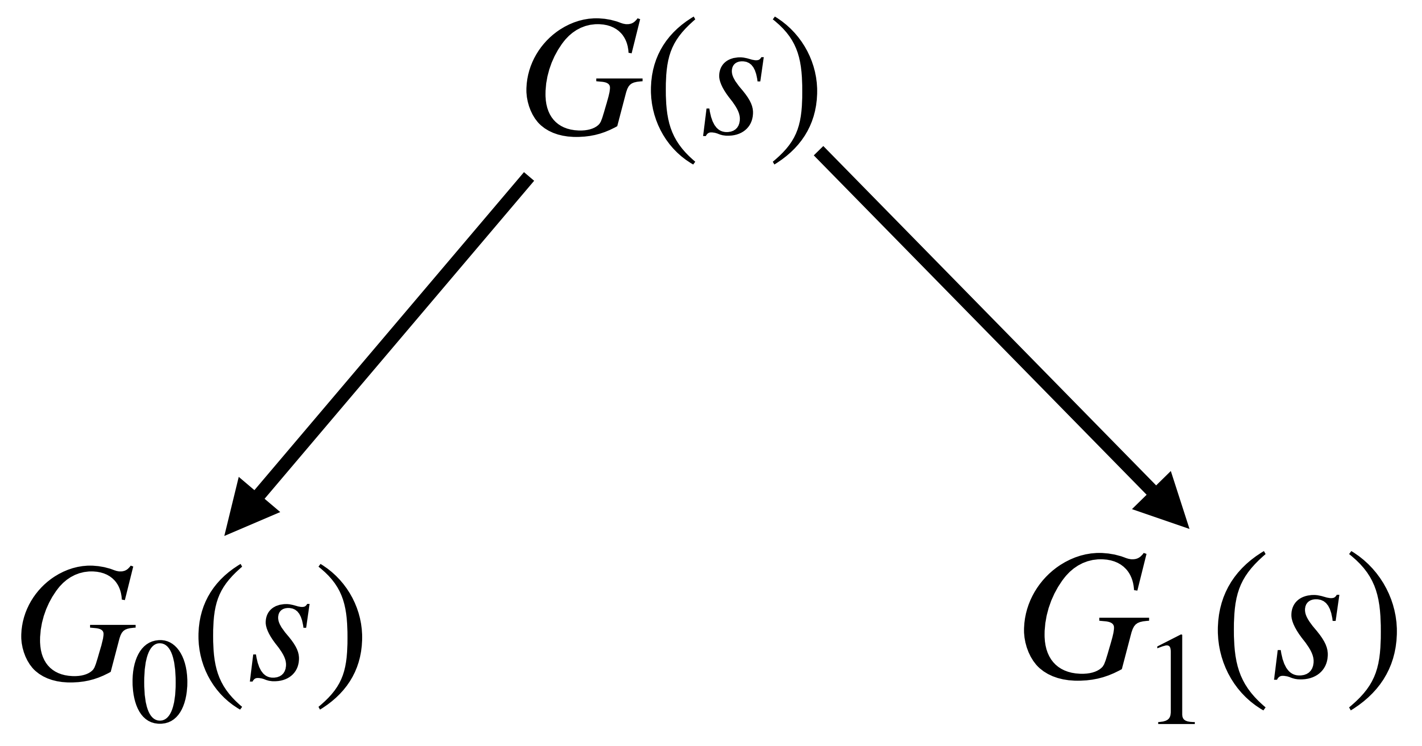
DPF construction

- Let $G : \{0,1\}^\lambda \rightarrow \{0,1\}^{2\lambda}$ be a length-doubling pseudorandom generator
- GGM-style pseudorandom function:
  - Given an output of $G(s)$, can divide into two halves where $G(s) = G_0(s) \,||\, G_1(s)$
  - If s is the root, then recursively applying G will result in a tree



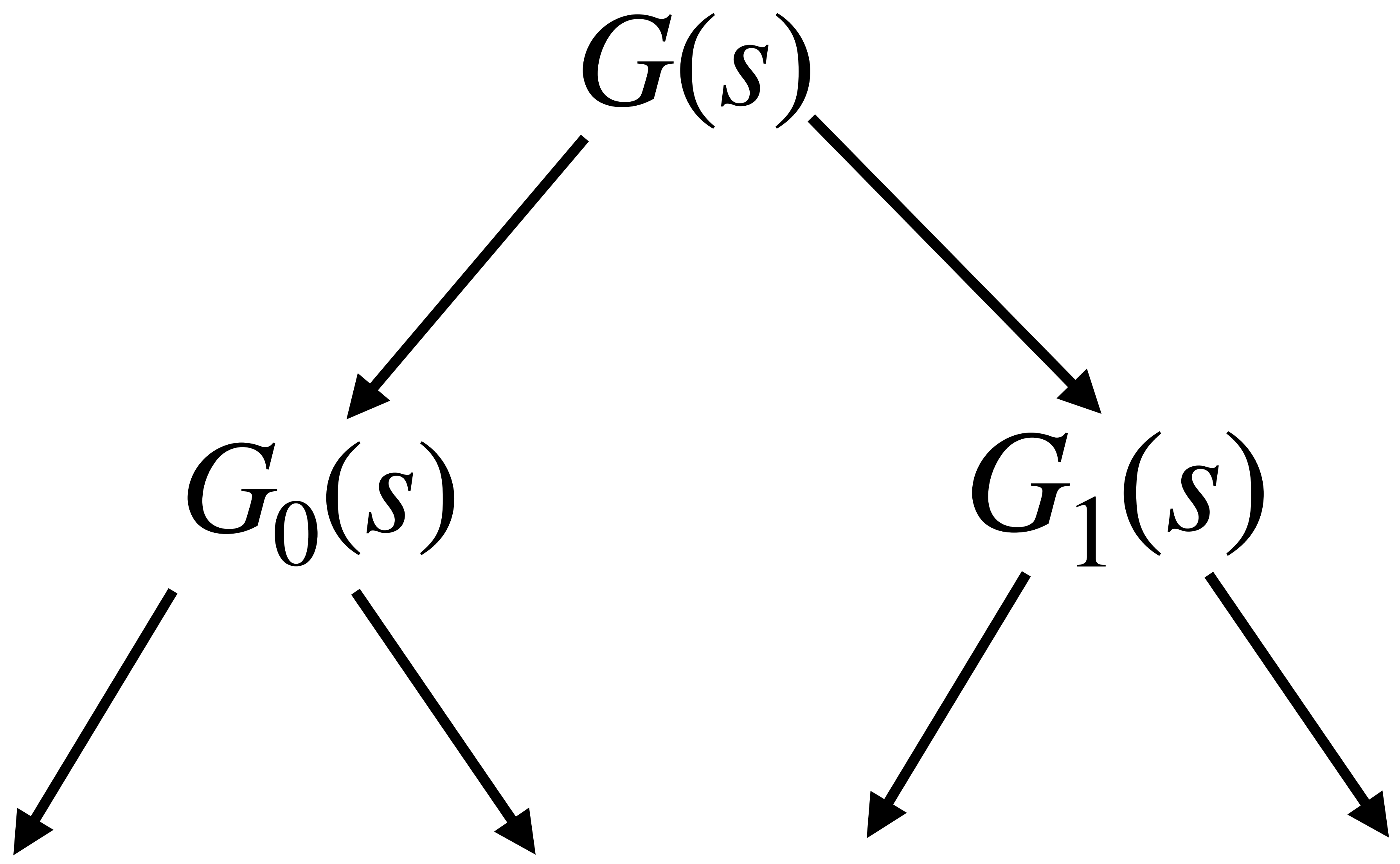
DPF construction

- Let $G : \{0,1\}^{\lambda} \rightarrow \{0,1\}^{2\lambda}$ be a length-doubling pseudorandom generator
- GGM-style pseudorandom function:
  - Given an output of $G(s)$, can divide into two halves where $G(s) = G_0(s) \,||\, G_1(s)$
  - If s is the root, then recursively applying G will result in a tree



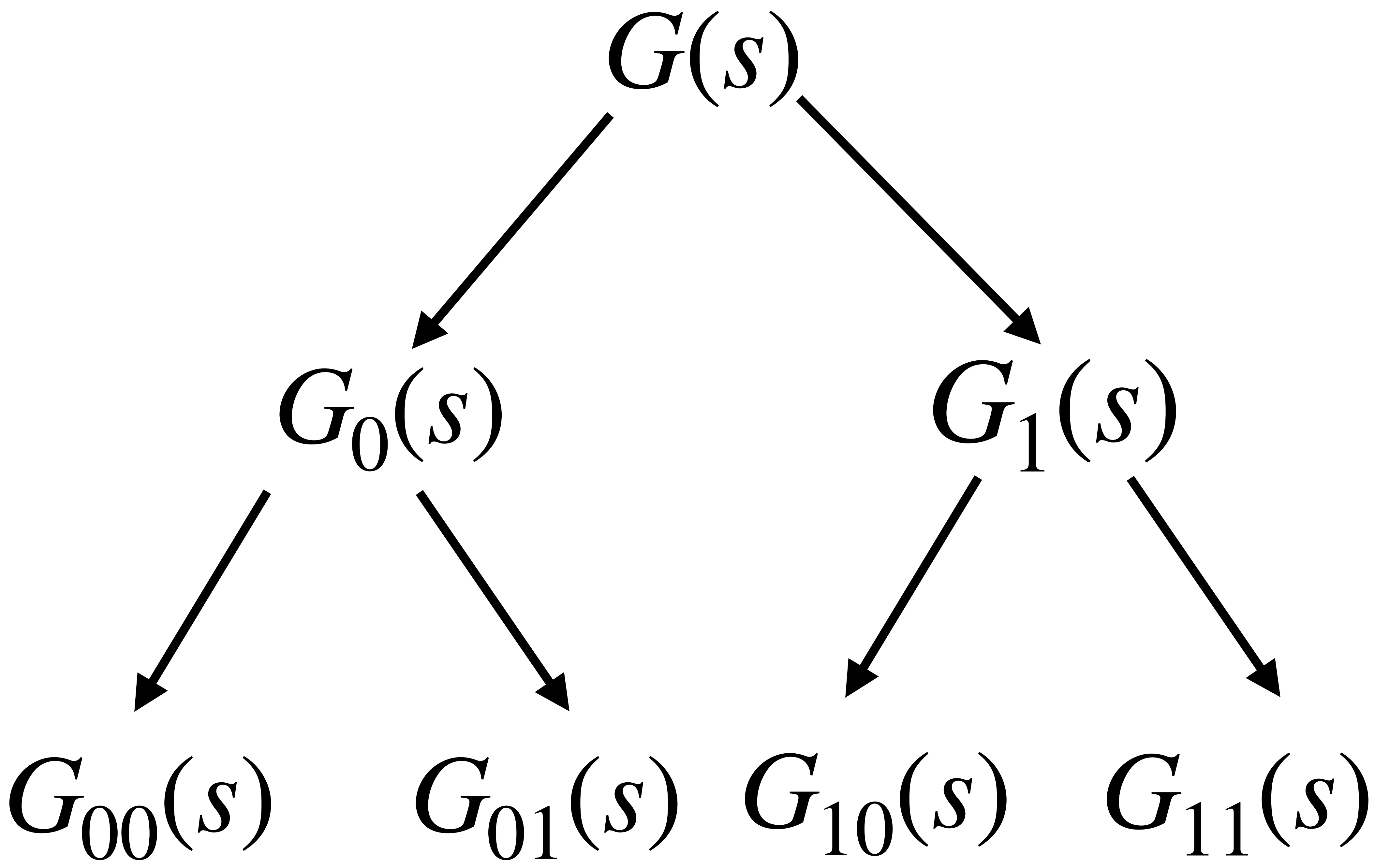
DPF construction

- Let $G : \{0,1\}^{\lambda} \rightarrow \{0,1\}^{2\lambda}$ be a length-doubling pseudorandom generator
- GGM-style pseudorandom function:
  - Given an output of $G(s)$, can divide into two halves where $G(s) = G_0(s) \,||\, G_1(s)$
  - If s is the root, then recursively applying G will result in a tree



DPF construction

- Let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ be a length-doubling pseudorandom generator
- GGM-style pseudorandom function:
  - Given an output of $G(s)$, can divide into two halves where $G(s) = G_0(s)\,||\,G_1(s)$
  - If s is the root, then recursively applying G will result in a tree



DPF construction

DPF construction

- A two-party, tree-based scheme with two keys, k_0, k_1

DPF construction

- A two-party, tree-based scheme with two keys, k_0, k_1
- Starting point is the GGM-style binary tree, but have a special evaluation path for the special input α

DPF construction

- A two-party, tree-based scheme with two keys, k_0, k_1
- Starting point is the GGM-style binary tree, but have a special evaluation path for the special input α
- Invariant for each node

DPF construction

- A two-party, tree-based scheme with two keys, k_0, k_1
- Starting point is the GGM-style binary tree, but have a special evaluation path for the special input α
- Invariant for each node
    - Each node has a seed, and a control bit

DPF construction

- A two-party, tree-based scheme with two keys, k_0, k_1
- Starting point is the GGM-style binary tree, but have a special evaluation path for the special input α
- Invariant for each node
    - Each node has a seed, and a control bit
    - Outside of special path: labels on the two trees are *identical*

DPF construction

- A two-party, tree-based scheme with two keys, k_0, k_1
- Starting point is the GGM-style binary tree, but have a special evaluation path for the special input α
- Invariant for each node
  - Each node has a seed, and a control bit
  - Outside of special path: labels on the two trees are *identical*
  - On the special path: seeds are indistinguishable from being random and independent, and two control bits are different

DPF construction

DPF construction

- Let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda+2}$ be a PRG, and let $[s]$ denote an additive secret sharing among two parties where the shares are s_0, s_1, and that $s = s_0 \oplus s_1$

DPF construction

- Let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda+2}$ be a PRG, and let $[s]$ denote an additive secret sharing among two parties where the shares are s_0, s_1, and that $s = s_0 \oplus s_1$
- Relies on two insights: properties of additive secret sharing specific to two parties

DPF construction

- Let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda+2}$ be a PRG, and let $[s]$ denote an additive secret sharing among two parties where the shares are s_0, s_1, and that $s = s_0 \oplus s_1$
- Relies on two insights: properties of additive secret sharing specific to two parties
  - **Weak homomorphism:** $G([s]) = (G(s_0), G(s_1))$ extends

DPF construction

- Let $G : \{0,1\}^{\lambda} \to \{0,1\}^{2\lambda+2}$ be a PRG, and let $[s]$ denote an additive secret sharing among two parties where the shares are s_0, s_1, and that $s = s_0 \oplus s_1$
- Relies on two insights: properties of additive secret sharing specific to two parties
  - **Weak homomorphism:** $G([s]) = (G(s_0), G(s_1))$ extends
    - Shares of the 0-string into shares of a longer 0-string. Why?

DPF construction

- Let $G : \{0,1\}^\lambda \rightarrow \{0,1\}^{2\lambda+2}$ be a PRG, and let $[s]$ denote an additive secret sharing among two parties where the shares are s_0, s_1, and that $s = s_0 \oplus s_1$
- Relies on two insights: properties of additive secret sharing specific to two parties
  - **Weak homomorphism:** $G([s]) = (G(s_0), G(s_1))$ extends
    - Shares of the 0-string into shares of a longer 0-string. Why? $s = 0 \rightarrow s_0 = s_1$

DPF construction

- Let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda+2}$ be a PRG, and let $[s]$ denote an additive secret sharing among two parties where the shares are s_0, s_1, and that $s = s_0 \oplus s_1$
- Relies on two insights: properties of additive secret sharing specific to two parties
  - **Weak homomorphism:** $G([s]) = (G(s_0), G(s_1))$ extends
    - Shares of the 0-string into shares of a longer 0-string. Why? $s = 0 \to s_0 = s_1$
    - Shares of a random seed s into shares of a longer pseudorandom string S

DPF construction

- Let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda+2}$ be a PRG, and let $[s]$ denote an additive secret sharing among two parties where the shares are s_0, s_1, and that $s = s_0 \oplus s_1$
- Relies on two insights: properties of additive secret sharing specific to two parties
  - **Weak homomorphism:** $G([s]) = (G(s_0), G(s_1))$ extends
    - Shares of the 0-string into shares of a longer 0-string. Why? $s = 0 \to s_0 = s_1$
    - Shares of a random seed s into shares of a longer pseudorandom string S
  - **Additive homomorphism:** given shares $[s], [t]$ of length λ and 1, respectively, and a public correction word CW, one can locally compute shares of $[s \oplus t \cdot CW]$

DPF construction

- Let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda+2}$ be a PRG, and let $[s]$ denote an additive secret sharing among two parties where the shares are s_0, s_1, and that $s = s_0 \oplus s_1$
- Relies on two insights: properties of additive secret sharing specific to two parties
  - **Weak homomorphism:** $G([s]) = (G(s_0), G(s_1))$ extends
    - Shares of the 0-string into shares of a longer 0-string. Why? $s = 0 \to s_0 = s_1$
    - Shares of a random seed s into shares of a longer pseudorandom string S
  - **Additive homomorphism:** given shares $[s], [t]$ of length λ and 1, respectively, and a public correction word CW, one can locally compute shares of $[s \oplus t \cdot CW]$ ***t's value chooses whether CW is applied to s***

DPF construction

$$\begin{tikzcd}[column sep=small] & \boxed{s_i^0, t_i^0} & \\ & G(s_i^0) = s_0^L, t_0^L, s_0^R, t_0^R \arrow[dl] \arrow[dr] & \\ \boxed{s_0^L, t_0^L} & & \boxed{s_0^R, t_0^R} \end{tikzcd}$$

$$\begin{tikzcd}[column sep=small] & \boxed{s_i^1, t_i^1} & \\ & G(s_i^1) = s_1^L, t_1^L, s_1^R, t_1^R \arrow[dl] \arrow[dr] & \\ \boxed{s_1^L, t_1^L} & & \boxed{s_1^R, t_1^R} \end{tikzcd}$$

DPF construction

s_i^0, t_i^0

s_i is seed, t_i is control bit

s_i^1, t_i^1

$G(s_i^0) = s_0^L, t_0^L, s_0^R, t_0^R$

$G(s_i^1) = s_1^L, t_1^L, s_1^R, t_1^R$

s_0^L, t_0^L

s_0^R, t_0^R

s_1^L, t_1^L

s_1^R, t_1^R

DPF construction

s_i **is seed,** t_i **is control bit**

$s_i = s_i^0 \oplus s_i^1, t_i = t_i^0 \oplus t_i^1$

$\boxed{s_i^0, t_i^0}$

$G(s_i^0) = s_0^L, t_0^L, s_0^R, t_0^R$

$\boxed{s_0^L, t_0^L}$ $\boxed{s_0^R, t_0^R}$

$\boxed{s_i^1, t_i^1}$

$G(s_i^1) = s_1^L, t_1^L, s_1^R, t_1^R$

$\boxed{s_1^L, t_1^L}$ $\boxed{s_1^R, t_1^R}$

DPF construction

s_i^0, t_i^0

s_i **is seed,** t_i **is control bit**

s_i^1, t_i^1

$s_i = s_i^0 \oplus s_i^1, t_i = t_i^0 \oplus t_i^1$

$G(s_i^0) = s_0^L, t_0^L, s_0^R, t_0^R$

$G(s_i^1) = s_1^L, t_1^L, s_1^R, t_1^R$

s_0^L, t_0^L $\quad$ s_0^R, t_0^R $\quad$ s_1^L, t_1^L $\quad$ s_1^R, t_1^R

If node is off the special path, then the weak homomorphism will allow expansion of the 0 string into a longer 0 string, maintaining identical labels

DPF construction

s_i^0, t_i^0

$$G(s_i^0) \oplus (t_0^i \cdot CW) = s_0^L, t_0^L, s_0^R, t_0^R$$

s_0^L, t_0^L

s_0^R, t_0^R

s_i^1, t_i^1

$$G(s_i^1) \oplus (t_1^i \cdot CW) = s_1^L, t_1^L, s_1^R, t_1^R$$

s_1^L, t_1^L

s_1^R, t_1^R

DPF construction

Because of additive homomorphism,
CW is only applied when the node is
on the special path!

$$
\begin{tikzcd}
 & s_i^0, t_i^0 \arrow[d] & \\
 & G(s_i^0) \oplus (t_0^i \cdot CW) = s_0^L, t_0^L, s_0^R, t_0^R \arrow[dl] \arrow[dr] & \\
s_0^L, t_0^L & & s_0^R, t_0^R
\end{tikzcd}
$$

$$
\begin{tikzcd}
 & s_i^1, t_i^1 \arrow[d] & \\
 & G(s_i^1) \oplus (t_1^i \cdot CW) = s_1^L, t_1^L, s_1^R, t_1^R \arrow[dl] \arrow[dr] & \\
s_1^L, t_1^L & & s_1^R, t_1^R
\end{tikzcd}
$$

DPF construction

s_i^0, t_i^0

$$G(s_i^0) \oplus (t_0^i \cdot CW) = s_0^L, t_0^L, s_0^R, t_0^R$$

s_0^L, t_0^L $\quad$ s_0^R, t_0^R

s_i^1, t_i^1

$$G(s_i^1) \oplus (t_1^i \cdot CW) = s_1^L, t_1^L, s_1^R, t_1^R$$

s_1^L, t_1^L $\quad$ s_1^R, t_1^R

DPF construction

s_i^0, t_i^0

s_i^1, t_i^1

$$G(s_i^0) \oplus (t_0^i \cdot CW) = s_0^L, t_0^L, s_0^R, t_0^R$$

$$G(s_i^1) \oplus (t_1^i \cdot CW) = s_1^L, t_1^L, s_1^R, t_1^R$$

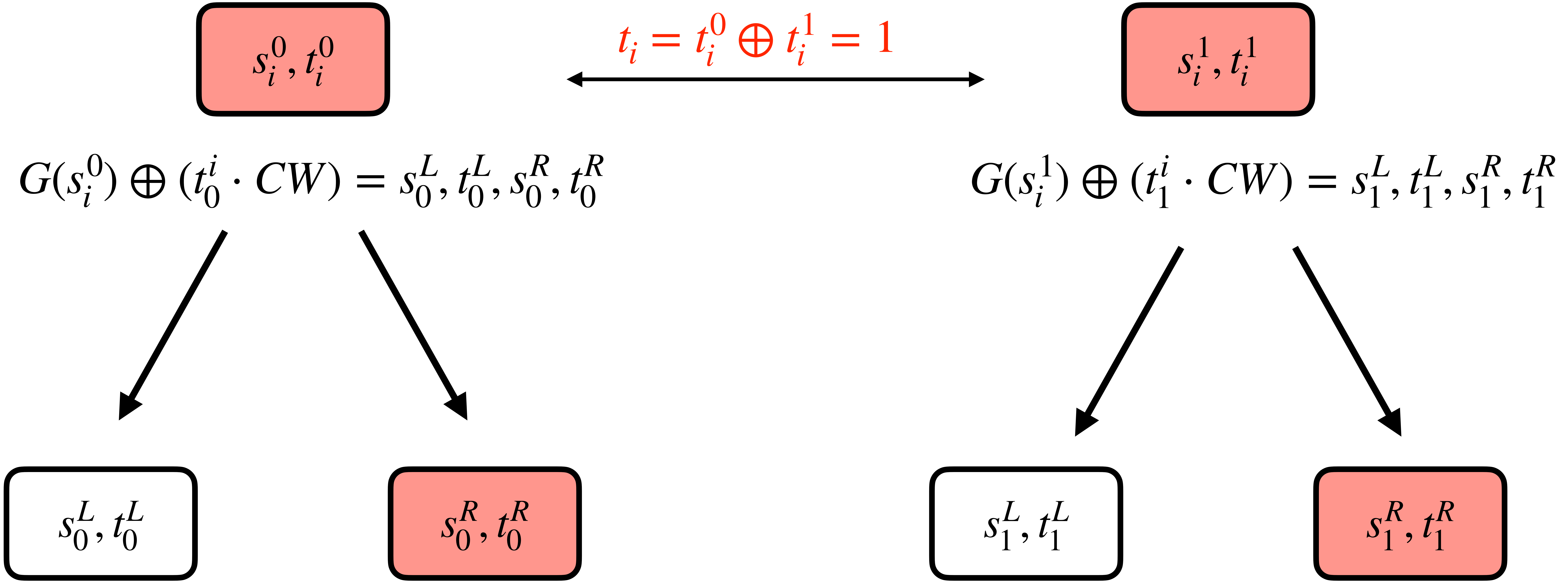
s_0^L, t_0^L

s_0^R, t_0^R

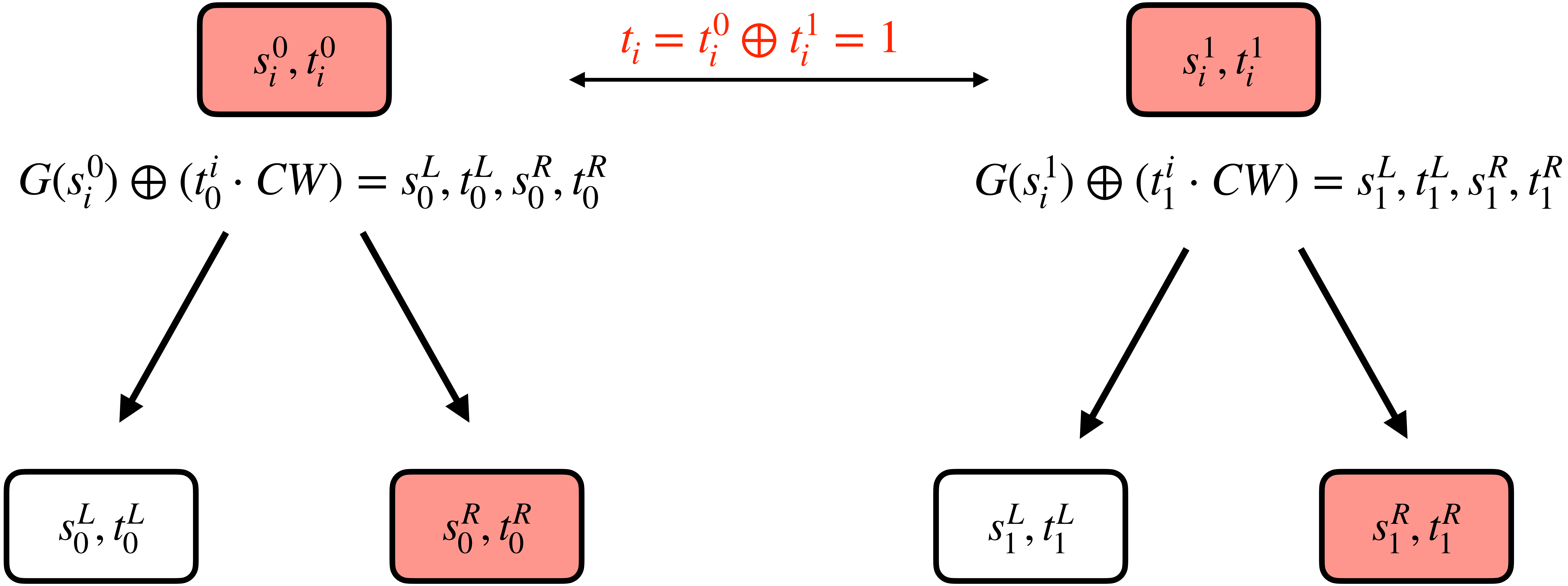
s_1^L, t_1^L

s_1^R, t_1^R

DPF construction

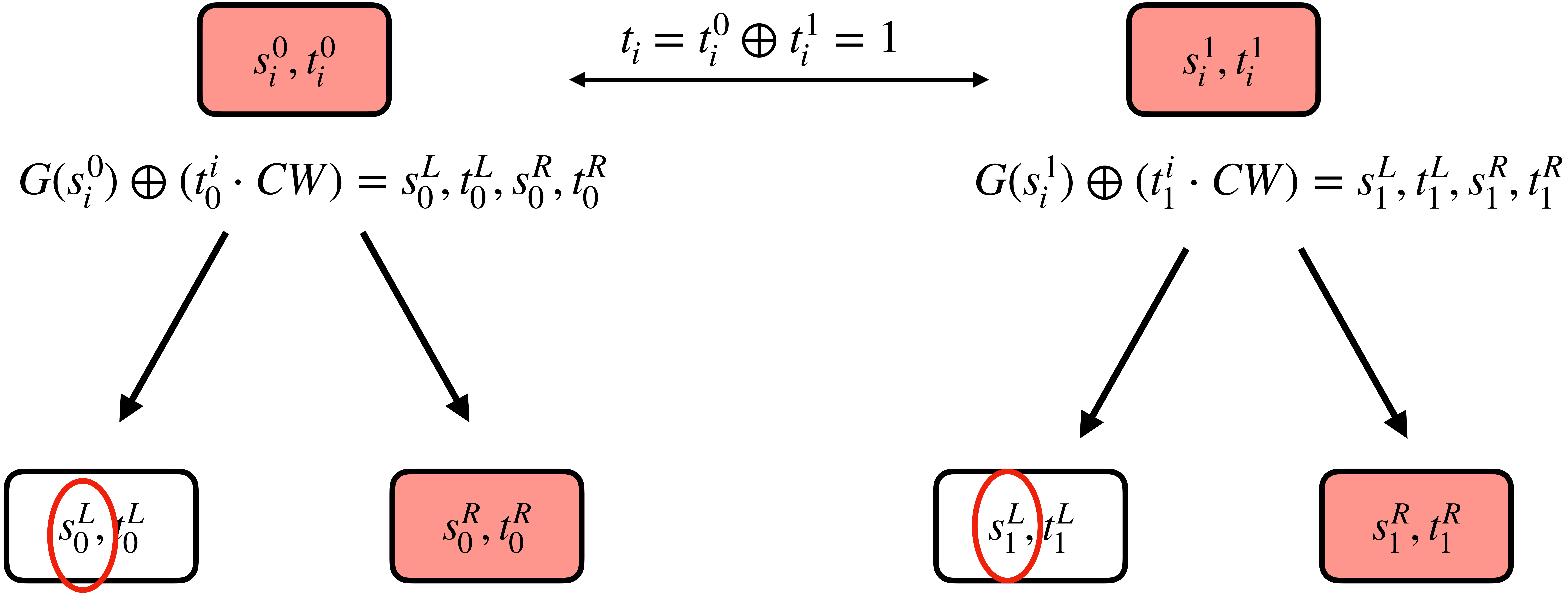


DPF construction



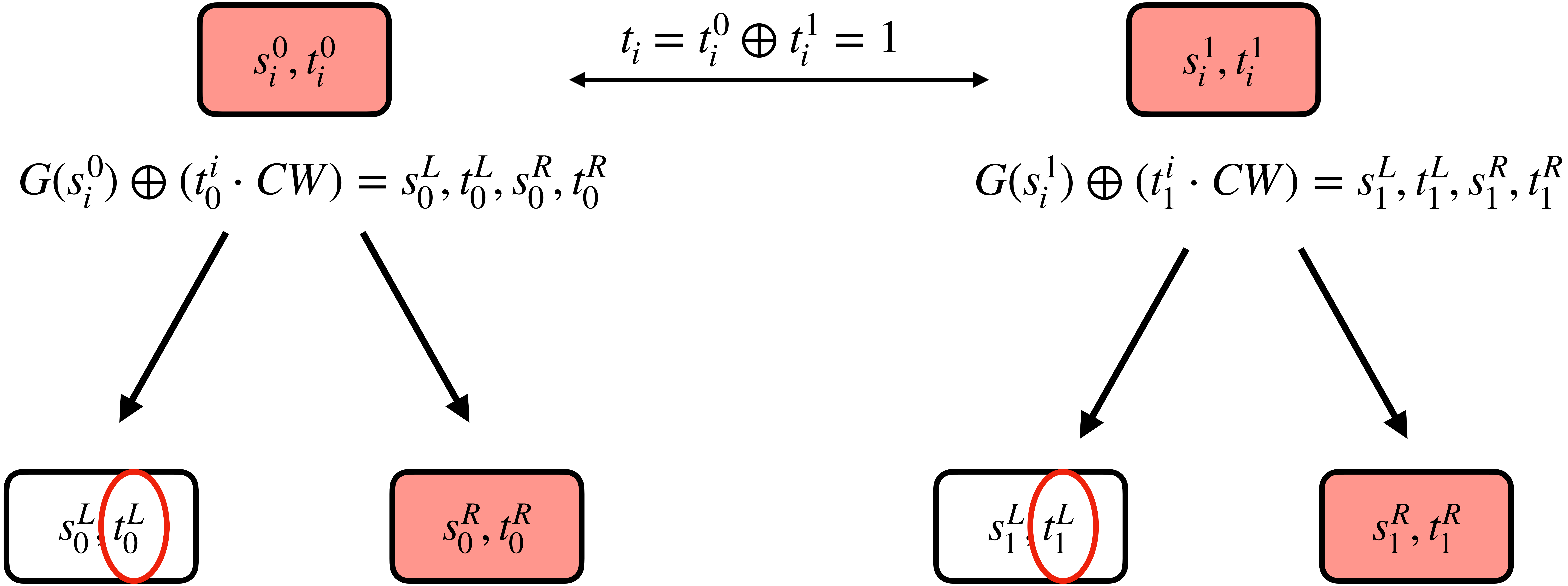
$CW =$

DPF construction



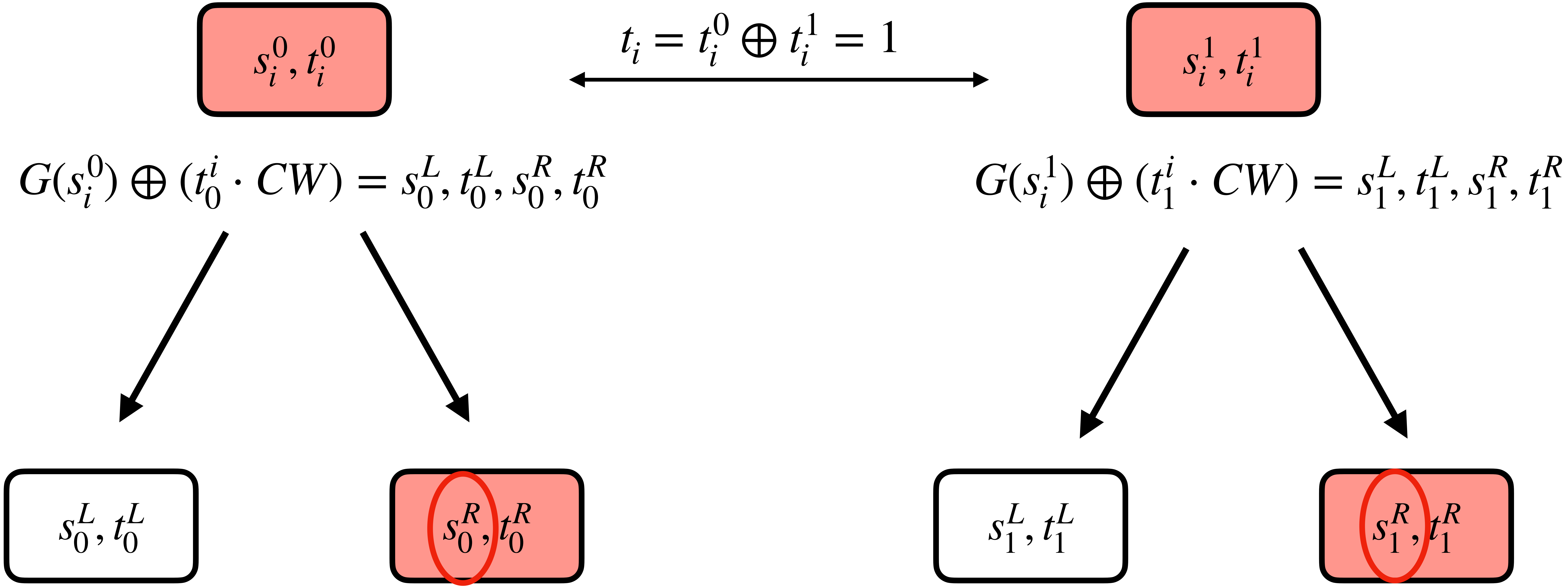
$$CW = s^L$$

DPF construction



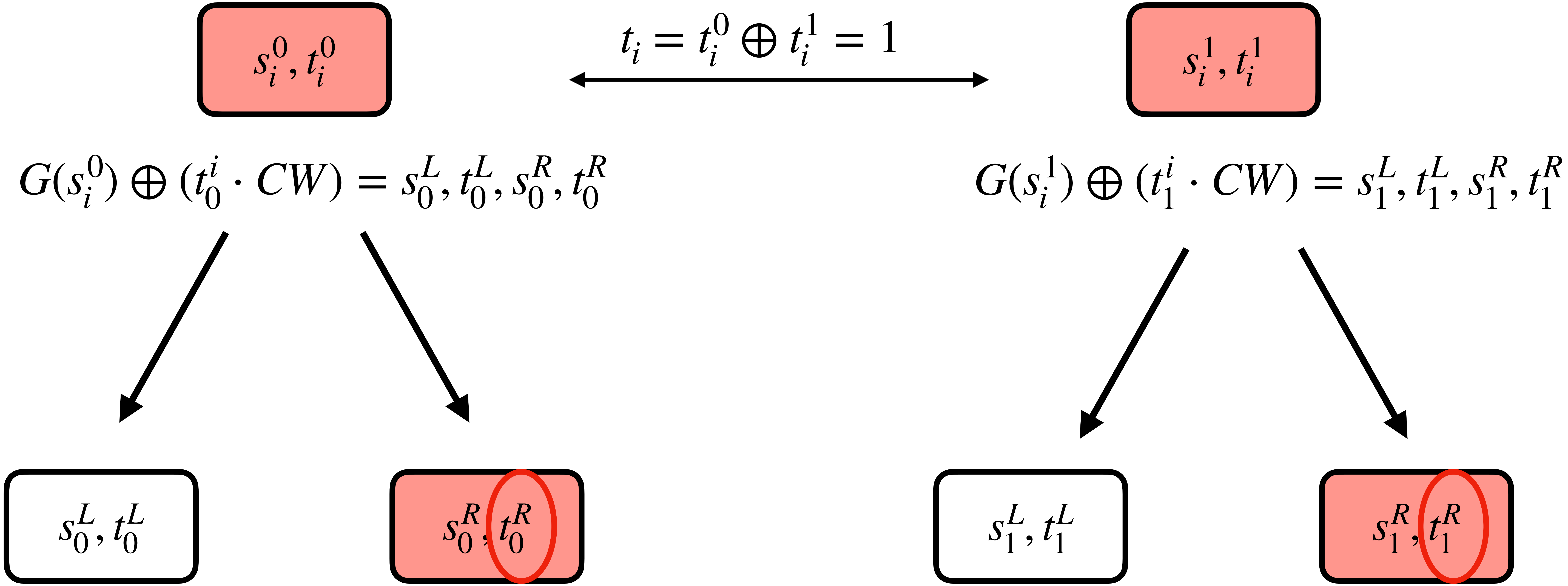
$$CW = s^L || t^L$$

DPF construction



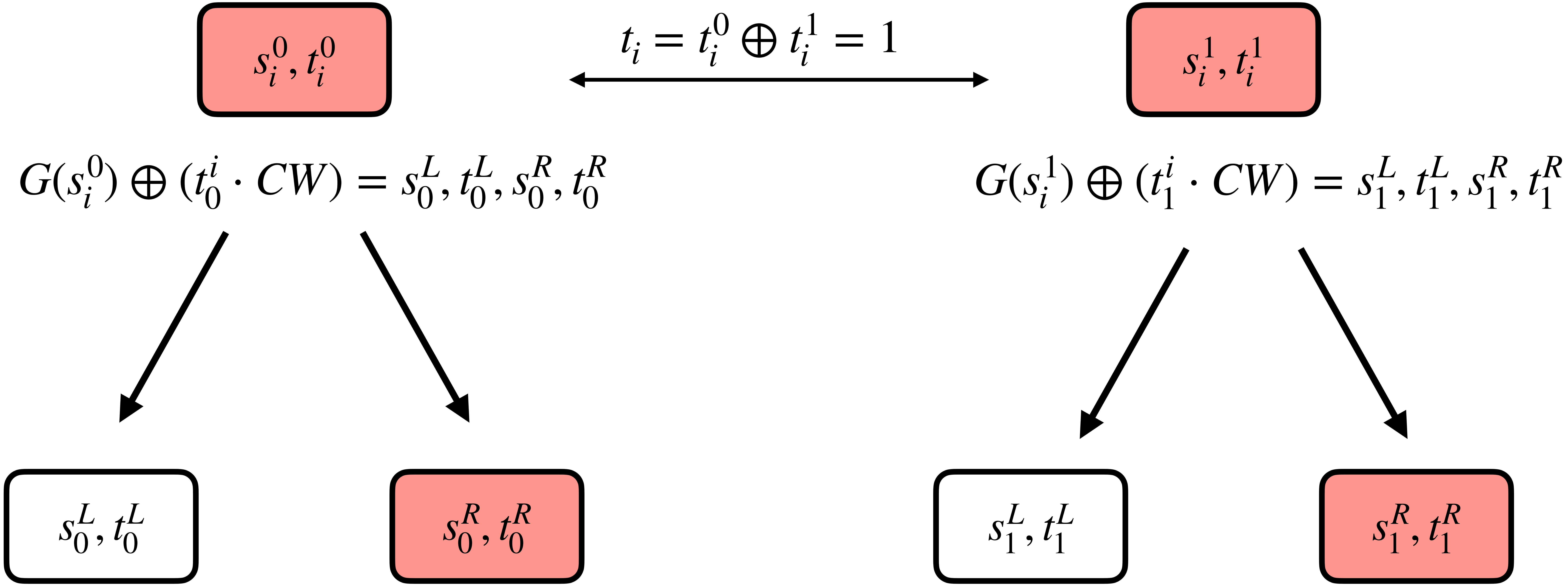
$$CW = s^L \,||\, t^L \,||\, s^R \oplus s'$$

DPF construction



$$CW = s^L \,||\, t^L \,||\, s^R \oplus s' \,||\, t^R \oplus 1$$

DPF construction



If reached leaf node, control bit t is either 0 or 1, and an additional CW is added to select β based on the value of the control bit

DPF construction

Optimized Distributed Point Function $(\mathsf{Gen}^\bullet, \mathsf{Eval}^\bullet)$
Let $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+1)}$ be a pseudorandom generator.
Let $\mathsf{Convert}_{\mathbb{G}} : \{0,1\}^\lambda \to \mathbb{G}$ be a map converting a random λ-bit string to a pseudorandom group element of $\mathbb{G}$. (See Figure 3.)

$\mathsf{Gen}^\bullet(1^\lambda, \alpha, \beta, \mathbb{G})$:

Let $\alpha = \alpha_1, \ldots, \alpha_n \in \{0,1\}^n$ be the bit decomposition of α
Sample random $s_0^{(0)} \leftarrow \{0,1\}^\lambda$ and $s_1^{(0)} \leftarrow \{0,1\}^\lambda$
Let $t_0^{(0)} = 0$ and $t_1^{(0)} = 1$
for $i = 1$ to n **do**
    $s_0^L || t_0^L \,||\, s_0^R || t_0^R \leftarrow G(s_0^{(i-1)})$ and $s_1^L || t_1^L \,||\, s_1^R || t_1^R \leftarrow G(s_1^{(i-1)})$.
    if $\alpha_i = 0$ **then** $\mathsf{Keep} \leftarrow L$, $\mathsf{Lose} \leftarrow R$
    else $\mathsf{Keep} \leftarrow R$, $\mathsf{Lose} \leftarrow L$
    end if
    $s_{CW} \leftarrow s_0^{\mathsf{Lose}} \oplus s_1^{\mathsf{Lose}}$
    $t_{CW}^L \leftarrow t_0^L \oplus t_1^L \oplus \alpha_i \oplus 1$ and $t_{CW}^R \leftarrow t_0^R \oplus t_1^R \oplus \alpha_i$
    $CW^{(i)} \leftarrow s_{CW} || t_{CW}^L || t_{CW}^R$
    $s_b^{(i)} \leftarrow s_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot s_{CW}$ for $b = 0, 1$
    $t_b^{(i)} \leftarrow t_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot t_{CW}^{\mathsf{Keep}}$ for $b = 0, 1$
end for
$CW^{(n+1)} \leftarrow (-1)^{t_1^n} \cdot \left[\beta - \mathsf{Convert}(s_0^{(n)}) + \mathsf{Convert}(s_1^{(n)})\right] \in \mathbb{G}$
Let $k_b = s_b^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$
return (k_0, k_1)

$\mathsf{Eval}^\bullet(b, k_b, x)$:

Parse $k_b = s^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$, and let $t^{(0)} = b$.
for $i = 1$ to n **do**
    Parse $CW^{(i)} = s_{CW} || t_{CW}^L || t_{CW}^R$
    $\tau^{(i)} \leftarrow G(s^{(i-1)}) \oplus (t^{(i-1)} \cdot \left[s_{CW} || t_{CW}^L || s_{CW} || t_{CW}^R\right])$
    Parse $\tau^{(i)} = s^L || t^L \,||\, s^R || t^R \in \{0,1\}^{2(\lambda+1)}$
    if $x_i = 0$ **then** $s^{(i)} \leftarrow s^L, t^{(i)} \leftarrow t^L$
    else $s^{(i)} \leftarrow s^R$, $t^{(i)} \leftarrow t^R$
    end if
end for
return $(-1)^b \cdot \left[\mathsf{Convert}(s^{(n)}) + t^{(n)} \cdot CW^{(n+1)}\right] \in \mathbb{G}$

DPF construction

Optimized Distributed Point Function $(\mathsf{Gen}^\bullet, \mathsf{Eval}^\bullet)$

Let $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+1)}$ be a pseudorandom generator.

Let $\mathsf{Convert}_{\mathbb{G}} : \{0,1\}^\lambda \to \mathbb{G}$ be a map converting a random λ-bit string to a pseudorandom group element of $\mathbb{G}$. (See Figure 3.)

$\mathsf{Gen}^\bullet(1^\lambda, \alpha, \beta, \mathbb{G})$:

Let $\alpha = \alpha_1, \ldots, \alpha_n \in \{0,1\}^n$ be the bit decomposition of α
Sample random $s_0^{(0)} \leftarrow \{0,1\}^\lambda$ and $s_1^{(0)} \leftarrow \{0,1\}^\lambda$
Let $t_0^{(0)} = 0$ and $t_1^{(0)} = 1$
for $i = 1$ to n **do**
    $s_0^L || t_0^L \;||\; s_0^R || t_0^R \leftarrow G(s_0^{(i-1)})$ and $s_1^L || t_1^L \;||\; s_1^R || t_1^R \leftarrow G(s_1^{(i-1)})$.
    if $\alpha_i = 0$ **then** $\mathsf{Keep} \leftarrow L$, $\mathsf{Lose} \leftarrow R$
    else $\mathsf{Keep} \leftarrow R$, $\mathsf{Lose} \leftarrow L$
    end if
    $s_{CW} \leftarrow s_0^{\mathsf{Lose}} \oplus s_1^{\mathsf{Lose}}$
    $t_{CW}^L \leftarrow t_0^L \oplus t_1^L \oplus \alpha_i \oplus 1$ and $t_{CW}^R \leftarrow t_0^R \oplus t_1^R \oplus \alpha_i$
    $CW^{(i)} \leftarrow s_{CW} || t_{CW}^L || t_{CW}^R$
    $s_b^{(i)} \leftarrow s_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot s_{CW}$ for $b = 0, 1$
    $t_b^{(i)} \leftarrow t_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot t_{CW}^{\mathsf{Keep}}$ for $b = 0, 1$
end for
$CW^{(n+1)} \leftarrow (-1)^{t_1^n} \cdot \left[\beta - \mathsf{Convert}(s_0^{(n)}) + \mathsf{Convert}(s_1^{(n)})\right] \in \mathbb{G}$
Let $k_b = s_b^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$
return (k_0, k_1)

$\mathsf{Eval}^\bullet(b, k_b, x)$:

Parse $k_b = s^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$, and let $t^{(0)} = b$.
for $i = 1$ to n **do**
    Parse $CW^{(i)} = s_{CW} || t_{CW}^L || t_{CW}^R$
    $\tau^{(i)} \leftarrow G(s^{(i-1)}) \oplus (t^{(i-1)} \cdot [s_{CW} || t_{CW}^L || s_{CW} || t_{CW}^R])$
    Parse $\tau^{(i)} = s^L || t^L \;||\; s^R || t^R \in \{0,1\}^{2(\lambda+1)}$
    if $x_i = 0$ **then** $s^{(i)} \leftarrow s^L, t^{(i)} \leftarrow t^L$
    else $s^{(i)} \leftarrow s^R$, $t^{(i)} \leftarrow t^R$
    end if
end for
return $(-1)^b \cdot \left[\mathsf{Convert}(s^{(n)}) + t^{(n)} \cdot CW^{(n+1)}\right] \in \mathbb{G}$

DPF construction

Optimized Distributed Point Function $(\mathsf{Gen}^\bullet, \mathsf{Eval}^\bullet)$

Let $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+1)}$ be a pseudorandom generator.

Let $\mathsf{Convert}_{\mathbb{G}} : \{0,1\}^\lambda \to \mathbb{G}$ be a map converting a random λ-bit string to a pseudorandom group element of $\mathbb{G}$. (See Figure 3.)

$\mathsf{Gen}^\bullet(1^\lambda, \alpha, \beta, \mathbb{G})$:

Let $\alpha = \alpha_1, \ldots, \alpha_n \in \{0,1\}^n$ be the bit decomposition of α
Sample random $s_0^{(0)} \leftarrow \{0,1\}^\lambda$ and $s_1^{(0)} \leftarrow \{0,1\}^\lambda$
Let $t_0^{(0)} = 0$ and $t_1^{(0)} = 1$
for $i = 1$ to n **do**
  $s_0^L || t_0^L \; || \; s_0^R || t_0^R \leftarrow G(s_0^{(i-1)})$ and $s_1^L || t_1^L \; || \; s_1^R || t_1^R \leftarrow G(s_1^{(i-1)})$.
  if $\alpha_i = 0$ **then** $\mathsf{Keep} \leftarrow L$, $\mathsf{Lose} \leftarrow R$
  else $\mathsf{Keep} \leftarrow R$, $\mathsf{Lose} \leftarrow L$
  end if
  $s_{CW} \leftarrow s_0^{\mathsf{Lose}} \oplus s_1^{\mathsf{Lose}}$
  $t_{CW}^L \leftarrow t_0^L \oplus t_1^L \oplus \alpha_i \oplus 1$ and $t_{CW}^R \leftarrow t_0^R \oplus t_1^R \oplus \alpha_i$
  $CW^{(i)} \leftarrow s_{CW} || t_{CW}^L || t_{CW}^R$
  $s_b^{(i)} \leftarrow s_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot s_{CW}$ for $b = 0, 1$
  $t_b^{(i)} \leftarrow t_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot t_{CW}^{\mathsf{Keep}}$ for $b = 0, 1$
end for
$CW^{(n+1)} \leftarrow (-1)^{t_1^n} \cdot \left[\beta - \mathsf{Convert}(s_0^{(n)}) + \mathsf{Convert}(s_1^{(n)})\right] \in \mathbb{G}$
Let $k_b = s_b^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$
return (k_0, k_1)

CW **generation**

$\mathsf{Eval}^\bullet(b, k_b, x)$:

Parse $k_b = s^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$, and let $t^{(0)} = b$.
for $i = 1$ to n **do**
  Parse $CW^{(i)} = s_{CW} || t_{CW}^L || t_{CW}^R$
  $\tau^{(i)} \leftarrow G(s^{(i-1)}) \oplus (t^{(i-1)} \cdot [s_{CW} || t_{CW}^L || s_{CW} || t_{CW}^R])$
  Parse $\tau^{(i)} = s^L || t^L \; || \; s^R || t^R \in \{0,1\}^{2(\lambda+1)}$
  if $x_i = 0$ **then** $s^{(i)} \leftarrow s^L$, $t^{(i)} \leftarrow t^L$
  else $s^{(i)} \leftarrow s^R$, $t^{(i)} \leftarrow t^R$
  end if
end for
return $(-1)^b \cdot \left[\mathsf{Convert}(s^{(n)}) + t^{(n)} \cdot CW^{(n+1)}\right] \in \mathbb{G}$

DPF construction

Optimized Distributed Point Function ($\mathsf{Gen}^\bullet, \mathsf{Eval}^\bullet$)

Let $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+1)}$ be a pseudorandom generator.

Let $\mathsf{Convert}_{\mathbb{G}} : \{0,1\}^\lambda \to \mathbb{G}$ be a map converting a random λ-bit string to a pseudorandom group element of $\mathbb{G}$. (See Figure 3.)

$\mathsf{Gen}^\bullet(1^\lambda, \alpha, \beta, \mathbb{G})$:

Let $\alpha = \alpha_1, \ldots, \alpha_n \in \{0,1\}^n$ be the bit decomposition of α

Sample random $s_0^{(0)} \leftarrow \{0,1\}^\lambda$ and $s_1^{(0)} \leftarrow \{0,1\}^\lambda$

Let $t_0^{(0)} = 0$ and $t_1^{(0)} = 1$

for $i = 1$ to n **do**

$\quad s_0^L || t_0^L \;||\; s_0^R || t_0^R \leftarrow G(s_0^{(i-1)})$ and $s_1^L || t_1^L \;||\; s_1^R || t_1^R \leftarrow G(s_1^{(i-1)})$.

$\quad$ **if** $\alpha_i = 0$ **then** $\mathsf{Keep} \leftarrow L$, $\mathsf{Lose} \leftarrow R$

$\quad$ **else** $\mathsf{Keep} \leftarrow R$, $\mathsf{Lose} \leftarrow L$

$\quad$ **end if**

$\quad s_{CW} \leftarrow s_0^{\mathsf{Lose}} \oplus s_1^{\mathsf{Lose}}$

$\quad t_{CW}^L \leftarrow t_0^L \oplus t_1^L \oplus \alpha_i \oplus 1$ and $t_{CW}^R \leftarrow t_0^R \oplus t_1^R \oplus \alpha_i$

$\quad CW^{(i)} \leftarrow s_{CW} || t_{CW}^L || t_{CW}^R$

CW **generation**

$\quad s_b^{(i)} \leftarrow s_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot s_{CW}$ for $b = 0, 1$

$\quad t_b^{(i)} \leftarrow t_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot t_{CW}^{\mathsf{Keep}}$ for $b = 0, 1$

end for

$CW^{(n+1)} \leftarrow (-1)^{t_1^{n}} \cdot \left[\beta - \mathsf{Convert}(s_0^{(n)}) + \mathsf{Convert}(s_1^{(n)})\right] \in \mathbb{G}$

Leaf *CW*

Let $k_b = s_b^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$

return (k_0, k_1)

$\mathsf{Eval}^\bullet(b, k_b, x)$:

Parse $k_b = s^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$, and let $t^{(0)} = b$.

for $i = 1$ to n **do**

$\quad$ Parse $CW^{(i)} = s_{CW} || t_{CW}^L || t_{CW}^R$

$\quad \tau^{(i)} \leftarrow G(s^{(i-1)}) \oplus (t^{(i-1)} \cdot [s_{CW} || t_{CW}^L || s_{CW} || t_{CW}^R])$

$\quad$ Parse $\tau^{(i)} = s^L || t^L \;||\; s^R || t^R \in \{0,1\}^{2(\lambda+1)}$

$\quad$ **if** $x_i = 0$ **then** $s^{(i)} \leftarrow s^L, t^{(i)} \leftarrow t^L$

$\quad$ **else** $s^{(i)} \leftarrow s^R$, $t^{(i)} \leftarrow t^R$

$\quad$ **end if**

end for

return $(-1)^b \cdot \left[\mathsf{Convert}(s^{(n)}) + t^{(n)} \cdot CW^{(n+1)}\right] \in \mathbb{G}$

DPF construction

Optimized Distributed Point Function $(\mathsf{Gen}^\bullet, \mathsf{Eval}^\bullet)$

Let $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+1)}$ be a pseudorandom generator.

Let $\mathsf{Convert}_{\mathbb{G}} : \{0,1\}^\lambda \to \mathbb{G}$ be a map converting a random λ-bit string to a pseudorandom group element of $\mathbb{G}$. (See Figure 3.)

$\mathsf{Gen}^\bullet(1^\lambda, \alpha, \beta, \mathbb{G})$:

Let $\alpha = \alpha_1, \ldots, \alpha_n \in \{0,1\}^n$ be the bit decomposition of α
Sample random $s_0^{(0)} \leftarrow \{0,1\}^\lambda$ and $s_1^{(0)} \leftarrow \{0,1\}^\lambda$
Let $t_0^{(0)} = 0$ and $t_1^{(0)} = 1$
for $i = 1$ to n **do**
 $s_0^L || t_0^L \mid\mid s_0^R || t_0^R \leftarrow G(s_0^{(i-1)})$ and $s_1^L || t_1^L \mid\mid s_1^R || t_1^R \leftarrow G(s_1^{(i-1)})$.
 if $\alpha_i = 0$ **then** $\mathsf{Keep} \leftarrow L$, $\mathsf{Lose} \leftarrow R$
 else $\mathsf{Keep} \leftarrow R$, $\mathsf{Lose} \leftarrow L$
 end if
 $s_{CW} \leftarrow s_0^{\mathsf{Lose}} \oplus s_1^{\mathsf{Lose}}$
 $t_{CW}^L \leftarrow t_0^L \oplus t_1^L \oplus \alpha_i \oplus 1$ and $t_{CW}^R \leftarrow t_0^R \oplus t_1^R \oplus \alpha_i$
 $CW^{(i)} \leftarrow s_{CW} || t_{CW}^L || t_{CW}^R$
 $s_b^{(i)} \leftarrow s_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot s_{CW}$ for $b = 0,1$
 $t_b^{(i)} \leftarrow t_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot t_{CW}^{\mathsf{Keep}}$ for $b = 0,1$
end for
$CW^{(n+1)} \leftarrow (-1)^{t_1^{(n)}} \cdot \left[\beta - \mathsf{Convert}(s_0^{(n)}) + \mathsf{Convert}(s_1^{(n)})\right] \in \mathbb{G}$
Let $k_b = s_b^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$
return (k_0, k_1)

CW **generation**

Leaf *CW*

$\mathsf{Eval}^\bullet(b, k_b, x)$:

Parse $k_b = s^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$, and let $t^{(0)} = b$.
for $i = 1$ to n **do**
 Parse $CW^{(i)} = s_{CW} || t_{CW}^L || t_{CW}^R$
 $\tau^{(i)} \leftarrow G(s^{(i-1)}) \oplus (t^{(i-1)} \cdot [s_{CW} || t_{CW}^L || s_{CW} || t_{CW}^R])$
 Parse $\tau^{(i)} = s^L || t^L \mid\mid s^R || t^R \in \{0,1\}^{2(\lambda+1)}$
 if $x_i = 0$ **then** $s^{(i)} \leftarrow s^L$, $t^{(i)} \leftarrow t^L$
 else $s^{(i)} \leftarrow s^R$, $t^{(i)} \leftarrow t^R$
 end if
end for
return $(-1)^b \cdot \left[\mathsf{Convert}(s^{(n)}) + t^{(n)} \cdot CW^{(n+1)}\right] \in \mathbb{G}$

DPF construction

Optimized Distributed Point Function $(\mathsf{Gen}^\bullet, \mathsf{Eval}^\bullet)$

Let $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+1)}$ be a pseudorandom generator.

Let $\mathsf{Convert}_{\mathbb{G}} : \{0,1\}^\lambda \to \mathbb{G}$ be a map converting a random λ-bit string to a pseudorandom group element of $\mathbb{G}$. (See Figure 3.)

$\mathsf{Gen}^\bullet(1^\lambda, \alpha, \beta, \mathbb{G})$:

```
Let α = α_1, ..., α_n ∈ {0,1}^n be the bit decomposition of α
Sample random s_0^(0) ← {0,1}^λ and s_1^(0) ← {0,1}^λ
Let t_0^(0) = 0 and t_1^(0) = 1
for i = 1 to n do
    s_0^L||t_0^L || s_0^R||t_0^R ← G(s_0^(i-1)) and s_1^L||t_1^L || s_1^R||t_1^R ← G(s_1^(i-1)).
    if α_i = 0 then Keep ← L, Lose ← R
    else Keep ← R, Lose ← L
    end if
    s_CW ← s_0^Lose ⊕ s_1^Lose
    t_CW^L ← t_0^L ⊕ t_1^L ⊕ α_i ⊕ 1 and t_CW^R ← t_0^R ⊕ t_1^R ⊕ α_i
    CW^(i) ← s_CW||t_CW^L||t_CW^R
    s_b^(i) ← s_b^Keep ⊕ t_b^(i-1) · s_CW for b = 0,1
    t_b^(i) ← t_b^Keep ⊕ t_b^(i-1) · t_CW^Keep for b = 0,1
end for
CW^(n+1) ← (-1)^(t_1^(n)) · [β - Convert(s_0^(n)) + Convert(s_1^(n))] ∈ G
Let k_b = s_b^(0)||CW^(1)||···||CW^(n+1)
return (k_0, k_1)
```

CW **generation**

Leaf CW

$\mathsf{Eval}^\bullet(b, k_b, x)$:

```
Parse k_b = s^(0)||CW^(1)||···||CW^(n+1), and let t^(0) = b.
for i = 1 to n do
    Parse CW^(i) = s_CW||t_CW^L||t_CW^R
    τ^(i) ← G(s^(i-1)) ⊕ (t^(i-1) · [s_CW||t_CW^L||s_CW||t_CW^R])
    Parse τ^(i) = s^L||t^L || s^R||t^R ∈ {0,1}^(2(λ+1))
    if x_i = 0 then s^(i) ← s^L, t^(i) ← t^L
    else s^(i) ← s^R, t^(i) ← t^R
    end if
end for
return (-1)^b · [Convert(s^(n)) + t^(n) · CW^(n+1)] ∈ G
```

DPF construction

Optimized Distributed Point Function ($\mathsf{Gen}^\bullet, \mathsf{Eval}^\bullet$)
Let $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+1)}$ be a pseudorandom generator.
Let $\mathsf{Convert}_{\mathbb{G}} : \{0,1\}^\lambda \to \mathbb{G}$ be a map converting a random λ-bit string to a pseudorandom group element of $\mathbb{G}$. (See Figure 3.)

$\mathsf{Gen}^\bullet(1^\lambda, \alpha, \beta, \mathbb{G})$:

Let $\alpha = \alpha_1, \dots, \alpha_n \in \{0,1\}^n$ be the bit decomposition of α
Sample random $s_0^{(0)} \leftarrow \{0,1\}^\lambda$ and $s_1^{(0)} \leftarrow \{0,1\}^\lambda$
Let $t_0^{(0)} = 0$ and $t_1^{(0)} = 1$
for $i = 1$ to n **do**
    $s_0^L || t_0^L \;||\; s_0^R || t_0^R \leftarrow G(s_0^{(i-1)})$ and $s_1^L || t_1^L \;||\; s_1^R || t_1^R \leftarrow G(s_1^{(i-1)})$.
    if $\alpha_i = 0$ **then** $\mathsf{Keep} \leftarrow L$, $\mathsf{Lose} \leftarrow R$
    else $\mathsf{Keep} \leftarrow R$, $\mathsf{Lose} \leftarrow L$
    end if
    $s_{CW} \leftarrow s_0^{\mathsf{Lose}} \oplus s_1^{\mathsf{Lose}}$
    $t_{CW}^L \leftarrow t_0^L \oplus t_1^L \oplus \alpha_i \oplus 1$ and $t_{CW}^R \leftarrow t_0^R \oplus t_1^R \oplus \alpha_i$ — CW **generation**
    $CW^{(i)} \leftarrow s_{CW} || t_{CW}^L || t_{CW}^R$
    $s_b^{(i)} \leftarrow s_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot s_{CW}$ for $b = 0, 1$
    $t_b^{(i)} \leftarrow t_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot t_{CW}^{\mathsf{Keep}}$ for $b = 0, 1$
end for
$CW^{(n+1)} \leftarrow (-1)^{t_1^{(n)}} \cdot \left[\beta - \mathsf{Convert}(s_0^{(n)}) + \mathsf{Convert}(s_1^{(n)})\right] \in \mathbb{G}$ — **Leaf** CW
Let $k_b = s_b^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$ — **Key handed to each party**
return (k_0, k_1)

$\mathsf{Eval}^\bullet(b, k_b, x)$:

Parse $k_b = s^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$, and let $t^{(0)} = b$.
for $i = 1$ to n **do**
    Parse $CW^{(i)} = s_{CW} || t_{CW}^L || t_{CW}^R$
    $\tau^{(i)} \leftarrow G(s^{(i-1)}) \oplus (t^{(i-1)} \cdot \left[s_{CW} || t_{CW}^L || s_{CW} || t_{CW}^R\right])$
    Parse $\tau^{(i)} = s^L || t^L \;||\; s^R || t^R \in \{0,1\}^{2(\lambda+1)}$
    if $x_i = 0$ **then** $s^{(i)} \leftarrow s^L$, $t^{(i)} \leftarrow t^L$
    else $s^{(i)} \leftarrow s^R$, $t^{(i)} \leftarrow t^R$
    end if
end for
return $(-1)^b \cdot \left[\mathsf{Convert}(s^{(n)}) + t^{(n)} \cdot CW^{(n+1)}\right] \in \mathbb{G}$

DPF construction

Optimized Distributed Point Function $(\mathsf{Gen}^\bullet, \mathsf{Eval}^\bullet)$

Let $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+1)}$ be a pseudorandom generator.

Let $\mathsf{Convert}_\mathbb{G} : \{0,1\}^\lambda \to \mathbb{G}$ be a map converting a random λ-bit string to a pseudorandom group element of $\mathbb{G}$. (See Figure 3.)

$\mathsf{Gen}^\bullet(1^\lambda, \alpha, \beta, \mathbb{G})$:

Let $\alpha = \alpha_1, \ldots, \alpha_n \in \{0,1\}^n$ be the bit decomposition of α
Sample random $s_0^{(0)} \leftarrow \{0,1\}^\lambda$ and $s_1^{(0)} \leftarrow \{0,1\}^\lambda$
Let $t_0^{(0)} = 0$ and $t_1^{(0)} = 1$
for $i = 1$ to n **do**
  $s_0^L||t_0^L \;||\; s_0^R||t_0^R \leftarrow G(s_0^{(i-1)})$ and $s_1^L||t_1^L \;||\; s_1^R||t_1^R \leftarrow G(s_1^{(i-1)})$.
  if $\alpha_i = 0$ **then** $\mathsf{Keep} \leftarrow L$, $\mathsf{Lose} \leftarrow R$
  else $\mathsf{Keep} \leftarrow R$, $\mathsf{Lose} \leftarrow L$
  end if
  $s_{CW} \leftarrow s_0^{\mathsf{Lose}} \oplus s_1^{\mathsf{Lose}}$
  $t_{CW}^L \leftarrow t_0^L \oplus t_1^L \oplus \alpha_i \oplus 1$ and $t_{CW}^R \leftarrow t_0^R \oplus t_1^R \oplus \alpha_i$
  $CW^{(i)} \leftarrow s_{CW}||t_{CW}^L||t_{CW}^R$

CW **generation**

  $s_b^{(i)} \leftarrow s_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot s_{CW}$ for $b = 0, 1$
  $t_b^{(i)} \leftarrow t_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot t_{CW}^{\mathsf{Keep}}$ for $b = 0, 1$
end for
$CW^{(n+1)} \leftarrow (-1)^{t_1^{(n)}} \cdot \left[\beta - \mathsf{Convert}(s_0^{(n)}) + \mathsf{Convert}(s_1^{(n)})\right] \in \mathbb{G}$

Leaf CW

Let $k_b = s_b^{(0)}||CW^{(1)}||\cdots||CW^{(n+1)}$

Key handed to each party

return (k_0, k_1)

$\mathsf{Eval}^\bullet(b, k_b, x)$:

Parse $k_b = s^{(0)}||CW^{(1)}||\cdots||CW^{(n+1)}$, and let $t^{(0)} = b$.
for $i = 1$ to n **do**
  Parse $CW^{(i)} = s_{CW}||t_{CW}^L||t_{CW}^R$
  $\tau^{(i)} \leftarrow G(s^{(i-1)}) \oplus (t^{(i-1)} \cdot [s_{CW}||t_{CW}^L||s_{CW}||t_{CW}^R])$
  Parse $\tau^{(i)} = s^L||t^L \;||\; s^R||t^R \in \{0,1\}^{2(\lambda+1)}$
  if $x_i = 0$ **then** $s^{(i)} \leftarrow s^L$, $t^{(i)} \leftarrow t^L$
  else $s^{(i)} \leftarrow s^R$, $t^{(i)} \leftarrow t^R$
  end if
end for
return $(-1)^b \cdot \left[\mathsf{Convert}(s^{(n)}) + t^{(n)} \cdot CW^{(n+1)}\right] \in \mathbb{G}$

Key size?

DPF construction

Optimized Distributed Point Function $(\mathsf{Gen}^\bullet, \mathsf{Eval}^\bullet)$
Let $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+1)}$ be a pseudorandom generator.
Let $\mathsf{Convert}_{\mathbb{G}} : \{0,1\}^\lambda \to \mathbb{G}$ be a map converting a random λ-bit string to a pseudorandom group element of $\mathbb{G}$. (See Figure 3.)

$\mathsf{Gen}^\bullet(1^\lambda, \alpha, \beta, \mathbb{G})$:

Let $\alpha = \alpha_1, \ldots, \alpha_n \in \{0,1\}^n$ be the bit decomposition of α
Sample random $s_0^{(0)} \leftarrow \{0,1\}^\lambda$ and $s_1^{(0)} \leftarrow \{0,1\}^\lambda$
Let $t_0^{(0)} = 0$ and $t_1^{(0)} = 1$
for $i = 1$ to n **do**
  $s_0^L || t_0^L \; || \; s_0^R || t_0^R \leftarrow G(s_0^{(i-1)})$ and $s_1^L || t_1^L \; || \; s_1^R || t_1^R \leftarrow G(s_1^{(i-1)})$.
  if $\alpha_i = 0$ **then** $\mathsf{Keep} \leftarrow L$, $\mathsf{Lose} \leftarrow R$
  else $\mathsf{Keep} \leftarrow R$, $\mathsf{Lose} \leftarrow L$
  end if
  $s_{CW} \leftarrow s_0^{\mathsf{Lose}} \oplus s_1^{\mathsf{Lose}}$
  $t_{CW}^L \leftarrow t_0^L \oplus t_1^L \oplus \alpha_i \oplus 1$ and $t_{CW}^R \leftarrow t_0^R \oplus t_1^R \oplus \alpha_i$
  $CW^{(i)} \leftarrow s_{CW} || t_{CW}^L || t_{CW}^R$

CW generation

  $s_b^{(i)} \leftarrow s_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot s_{CW}$ for $b = 0, 1$
  $t_b^{(i)} \leftarrow t_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot t_{CW}^{\mathsf{Keep}}$ for $b = 0, 1$
end for
$CW^{(n+1)} \leftarrow (-1)^{t_1^{(n)}} \cdot \left[\beta - \mathsf{Convert}(s_0^{(n)}) + \mathsf{Convert}(s_1^{(n)})\right] \in \mathbb{G}$

Leaf CW

Let $k_b = s_b^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$

Key handed to each party

return (k_0, k_1)

$\mathsf{Eval}^\bullet(b, k_b, x)$:

Parse $k_b = s^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$, and let $t^{(0)} = b$.
for $i = 1$ to n **do**
  Parse $CW^{(i)} = s_{CW} || t_{CW}^L || t_{CW}^R$
  $\tau^{(i)} \leftarrow G(s^{(i-1)}) \oplus (t^{(i-1)} \cdot [s_{CW} || t_{CW}^L || s_{CW} || t_{CW}^R])$
  Parse $\tau^{(i)} = s^L || t^L \; || \; s^R || t^R \in \{0,1\}^{2(\lambda+1)}$
  if $x_i = 0$ **then** $s^{(i)} \leftarrow s^L$, $t^{(i)} \leftarrow t^L$
  else $s^{(i)} \leftarrow s^R$, $t^{(i)} \leftarrow t^R$
  end if
end for
return $(-1)^b \cdot \left[\mathsf{Convert}(s^{(n)}) + t^{(n)} \cdot CW^{(n+1)}\right] \in \mathbb{G}$

Key size?

$O(\lambda n)$ where n is the number of bits of input

DPF construction

Optimized Distributed Point Function $(\mathsf{Gen}^\bullet, \mathsf{Eval}^\bullet)$

Let $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+1)}$ be a pseudorandom generator.

Let $\mathsf{Convert}_\mathbb{G} : \{0,1\}^\lambda \to \mathbb{G}$ be a map converting a random λ-bit string to a pseudorandom group element of $\mathbb{G}$. (See Figure 3.)

$\mathsf{Gen}^\bullet(1^\lambda, \alpha, \beta, \mathbb{G})$:

Let $\alpha = \alpha_1, \ldots, \alpha_n \in \{0,1\}^n$ be the bit decomposition of α
Sample random $s_0^{(0)} \leftarrow \{0,1\}^\lambda$ and $s_1^{(0)} \leftarrow \{0,1\}^\lambda$
Let $t_0^{(0)} = 0$ and $t_1^{(0)} = 1$
for $i = 1$ to n **do**
  $s_0^L || t_0^L \;||\; s_0^R || t_0^R \leftarrow G(s_0^{(i-1)})$ and $s_1^L || t_1^L \;||\; s_1^R || t_1^R \leftarrow G(s_1^{(i-1)})$.
  if $\alpha_i = 0$ **then** $\mathsf{Keep} \leftarrow L$, $\mathsf{Lose} \leftarrow R$
  else $\mathsf{Keep} \leftarrow R$, $\mathsf{Lose} \leftarrow L$
  end if
  $s_{CW} \leftarrow s_0^{\mathsf{Lose}} \oplus s_1^{\mathsf{Lose}}$
  $t_{CW}^L \leftarrow t_0^L \oplus t_1^L \oplus \alpha_i \oplus 1$ and $t_{CW}^R \leftarrow t_0^R \oplus t_1^R \oplus \alpha_i$
  $CW^{(i)} \leftarrow s_{CW} || t_{CW}^L || t_{CW}^R$
  $s_b^{(i)} \leftarrow s_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot s_{CW}$ for $b = 0, 1$
  $t_b^{(i)} \leftarrow t_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot t_{CW}^{\mathsf{Keep}}$ for $b = 0, 1$
end for
$CW^{(n+1)} \leftarrow (-1)^{t_1^n} \cdot \left[\beta - \mathsf{Convert}(s_0^{(n)}) + \mathsf{Convert}(s_1^{(n)})\right] \in \mathbb{G}$
Let $k_b = s_b^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$
return (k_0, k_1)

CW **generation**

Leaf CW

Key handed to each party

$\mathsf{Eval}^\bullet(b, k_b, x)$:

Parse $k_b = s^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$, and let $t^{(0)} = b$.
for $i = 1$ to n **do**
  Parse $CW^{(i)} = s_{CW} || t_{CW}^L || t_{CW}^R$
  $\tau^{(i)} \leftarrow G(s^{(i-1)}) \oplus (t^{(i-1)} \cdot [s_{CW} || t_{CW}^L || s_{CW} || t_{CW}^R])$
  Parse $\tau^{(i)} = s^L || t^L \;||\; s^R || t^R \in \{0,1\}^{2(\lambda+1)}$
  if $x_i = 0$ **then** $s^{(i)} \leftarrow s^L, t^{(i)} \leftarrow t^L$
  else $s^{(i)} \leftarrow s^R$, $t^{(i)} \leftarrow t^R$
  end if
end for
return $(-1)^b \cdot \left[\mathsf{Convert}(s^{(n)}) + t^{(n)} \cdot CW^{(n+1)}\right] \in \mathbb{G}$

Key size?

$O(\lambda n)$ **where** n **is the number of bits of input**

DPF construction

Optimized Distributed Point Function ($\mathsf{Gen}^\bullet, \mathsf{Eval}^\bullet$)

Let $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+1)}$ be a pseudorandom generator.

Let $\mathsf{Convert}_{\mathbb{G}} : \{0,1\}^\lambda \to \mathbb{G}$ be a map converting a random λ-bit string to a pseudorandom group element of $\mathbb{G}$. (See Figure 3.)

$\mathsf{Gen}^\bullet(1^\lambda, \alpha, \beta, \mathbb{G})$:

Let $\alpha = \alpha_1, \ldots, \alpha_n \in \{0,1\}^n$ be the bit decomposition of α
Sample random $s_0^{(0)} \leftarrow \{0,1\}^\lambda$ and $s_1^{(0)} \leftarrow \{0,1\}^\lambda$
Let $t_0^{(0)} = 0$ and $t_1^{(0)} = 1$
for $i = 1$ to n **do**
  $s_0^L || t_0^L \; || \; s_0^R || t_0^R \leftarrow G(s_0^{(i-1)})$ and $s_1^L || t_1^L \; || \; s_1^R || t_1^R \leftarrow G(s_1^{(i-1)})$.
  if $\alpha_i = 0$ **then** $\mathsf{Keep} \leftarrow L$, $\mathsf{Lose} \leftarrow R$
  else $\mathsf{Keep} \leftarrow R$, $\mathsf{Lose} \leftarrow L$
  end if
  $s_{CW} \leftarrow s_0^{\mathsf{Lose}} \oplus s_1^{\mathsf{Lose}}$
  $t_{CW}^L \leftarrow t_0^L \oplus t_1^L \oplus \alpha_i \oplus 1$ and $t_{CW}^R \leftarrow t_0^R \oplus t_1^R \oplus \alpha_i$
  $CW^{(i)} \leftarrow s_{CW} || t_{CW}^L || t_{CW}^R$
  $s_b^{(i)} \leftarrow s_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot s_{CW}$ for $b = 0, 1$
  $t_b^{(i)} \leftarrow t_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot t_{CW}^{\mathsf{Keep}}$ for $b = 0, 1$
end for
$CW^{(n+1)} \leftarrow (-1)^{t_1^{n}} \cdot \left[\beta - \mathsf{Convert}(s_0^{(n)}) + \mathsf{Convert}(s_1^{(n)})\right] \in \mathbb{G}$
Let $k_b = s_b^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$
return (k_0, k_1)

CW **generation**

Leaf CW

Key handed to each party

$\mathsf{Eval}^\bullet(b, k_b, x)$:

Parse $k_b = s^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$, and let $t^{(0)} = b$.
for $i = 1$ to n **do**
  Parse $CW^{(i)} = s_{CW} || t_{CW}^L || t_{CW}^R$
  $\tau^{(i)} \leftarrow G(s^{(i-1)}) \oplus (t^{(i-1)} \cdot [s_{CW} || t_{CW}^L || s_{CW} || t_{CW}^R])$
  Parse $\tau^{(i)} = s^L || t^L \; || \; s^R || t^R \in \{0,1\}^{2(\lambda+1)}$
  if $x_i = 0$ **then** $s^{(i)} \leftarrow s^L, t^{(i)} \leftarrow t^L$
  else $s^{(i)} \leftarrow s^R$, $t^{(i)} \leftarrow t^R$
  end if
end for
return $(-1)^b \cdot \left[\mathsf{Convert}(s^{(n)}) + t^{(n)} \cdot CW^{(n+1)}\right] \in \mathbb{G}$

Key size?

$O(\lambda n)$ **where** n **is the number of bits of input**

Security?

DPF construction

Optimized Distributed Point Function $(\mathsf{Gen}^\bullet, \mathsf{Eval}^\bullet)$

Let $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+1)}$ be a pseudorandom generator.

Let $\mathsf{Convert}_{\mathbb{G}} : \{0,1\}^\lambda \to \mathbb{G}$ be a map converting a random λ-bit string to a pseudorandom group element of $\mathbb{G}$. (See Figure 3.)

$\mathsf{Gen}^\bullet(1^\lambda, \alpha, \beta, \mathbb{G})$:

Let $\alpha = \alpha_1, \ldots, \alpha_n \in \{0,1\}^n$ be the bit decomposition of α

Sample random $s_0^{(0)} \leftarrow \{0,1\}^\lambda$ and $s_1^{(0)} \leftarrow \{0,1\}^\lambda$

Let $t_0^{(0)} = 0$ and $t_1^{(0)} = 1$

for $i = 1$ to n **do**

$\quad s_0^L || t_0^L \; || \; s_0^R || t_0^R \leftarrow G(s_0^{(i-1)})$ and $s_1^L || t_1^L \; || \; s_1^R || t_1^R \leftarrow G(s_1^{(i-1)})$.

$\quad$ **if** $\alpha_i = 0$ **then** $\mathsf{Keep} \leftarrow L$, $\mathsf{Lose} \leftarrow R$

$\quad$ **else** $\mathsf{Keep} \leftarrow R$, $\mathsf{Lose} \leftarrow L$

$\quad$ **end if**

$\quad s_{CW} \leftarrow s_0^{\mathsf{Lose}} \oplus s_1^{\mathsf{Lose}}$

$\quad t_{CW}^L \leftarrow t_0^L \oplus t_1^L \oplus \alpha_i \oplus 1$ and $t_{CW}^R \leftarrow t_0^R \oplus t_1^R \oplus \alpha_i$

$\quad CW^{(i)} \leftarrow s_{CW} || t_{CW}^L || t_{CW}^R$

CW **generation**

$\quad s_b^{(i)} \leftarrow s_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot s_{CW}$ for $b = 0, 1$

$\quad t_b^{(i)} \leftarrow t_b^{\mathsf{Keep}} \oplus t_b^{(i-1)} \cdot t_{CW}^{\mathsf{Keep}}$ for $b = 0, 1$

end for

$CW^{(n+1)} \leftarrow (-1)^{t_1^{(n)}} \cdot \left[\beta - \mathsf{Convert}(s_0^{(n)}) + \mathsf{Convert}(s_1^{(n)})\right] \in \mathbb{G}$

Leaf *CW*

Let $k_b = s_b^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$

Key handed to each party

return (k_0, k_1)

$\mathsf{Eval}^\bullet(b, k_b, x)$:

Parse $k_b = s^{(0)} || CW^{(1)} || \cdots || CW^{(n+1)}$, and let $t^{(0)} = b$.

for $i = 1$ to n **do**

$\quad$ Parse $CW^{(i)} = s_{CW} || t_{CW}^L || t_{CW}^R$

$\quad \tau^{(i)} \leftarrow G(s^{(i-1)}) \oplus (t^{(i-1)} \cdot [s_{CW} || t_{CW}^L || s_{CW} || t_{CW}^R])$

$\quad$ Parse $\tau^{(i)} = s^L || t^L \; || \; s^R || t^R \in \{0,1\}^{2(\lambda+1)}$

$\quad$ **if** $x_i = 0$ **then** $s^{(i)} \leftarrow s^L$, $t^{(i)} \leftarrow t^L$

$\quad$ **else** $s^{(i)} \leftarrow s^R$, $t^{(i)} \leftarrow t^R$

$\quad$ **end if**

end for

return $(-1)^b \cdot \left[\mathsf{Convert}(s^{(n)}) + t^{(n)} \cdot CW^{(n+1)}\right] \in \mathbb{G}$

Key size?

$O(\lambda n)$ **where** n **is the number of bits of input**

Security?

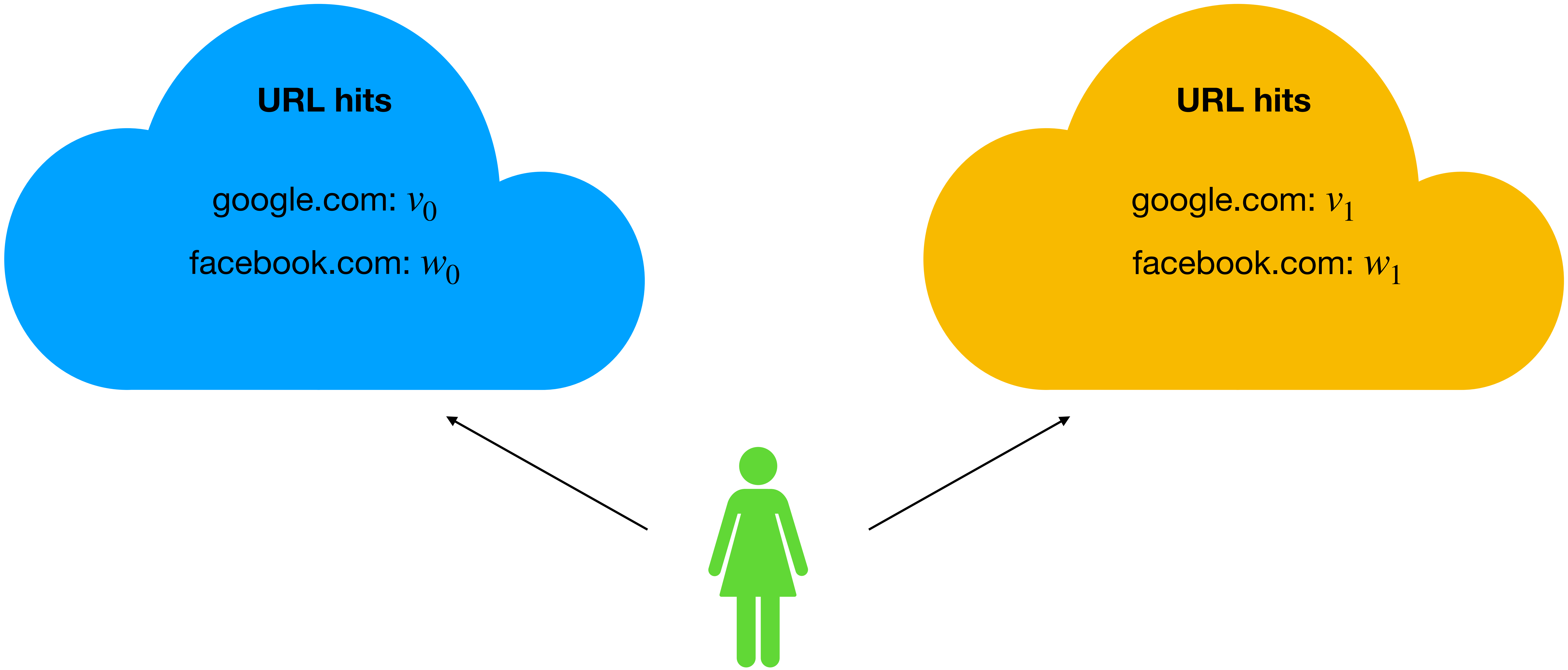
k_b **is pseudorandom because 1) seed is random 2) CW use up 3/4 generated randomness**

Applications of DPF

- Private keyword search

Applications of DPF

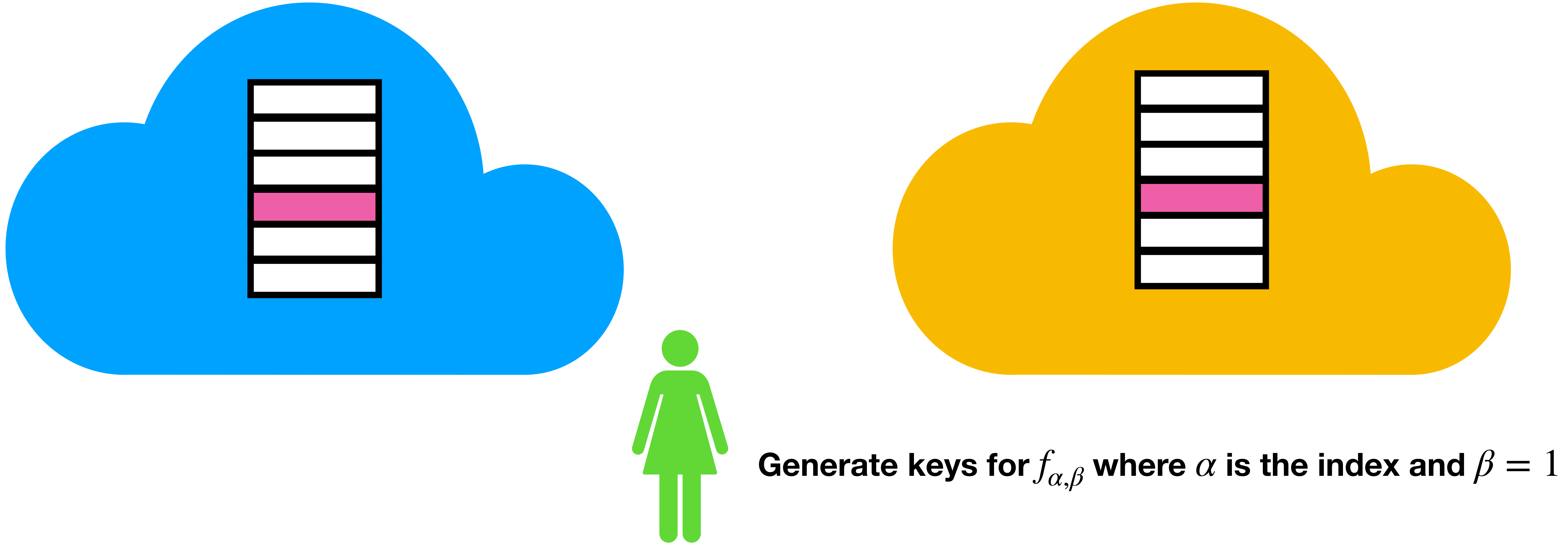
- Private statistics collection



Today: anonymous messaging

Next time: Pung

- DPF can be used for Private Information Retrieval (PIR)
    - Allows clients to fetch item i from a database of n items without revealing i



Next time: Pung

Next time: Pung

- One weakness of DPF: requires non-colluding servers

Next time: Pung

- One weakness of DPF: requires non-colluding servers
- Is it possible to only use a single server that’s fully untrusted?

Next time: Pung

- One weakness of DPF: requires non-colluding servers
- Is it possible to only use a single server that's fully untrusted?
  - Single server computational private information retrieval

Next time: Pung

- One weakness of DPF: requires non-colluding servers
- Is it possible to only use a single server that's fully untrusted?
  - Single server computational private information retrieval
- Is it possible to reduce the cost of a retrieval?

Next time: Pung

- One weakness of DPF: requires non-colluding servers
- Is it possible to only use a single server that's fully untrusted?
  - Single server computational private information retrieval
- Is it possible to reduce the cost of a retrieval?
  - Batching queries together for better throughput