
Algorithm 1 INVERT-BIT($[b]^i$)

```
1: if  $i = n \wedge n$  is even then  
2:   return  $[b]^i$   
3: else  
4:   return  $\neg[b]^i$   
5: end if
```

Algorithm 2 R-CONT($[c], [d_i]$)

```
1:  $A \leftarrow$  Array of size  $l - r + 1$   
2: for  $f = r$  to  $l$  do  
3:    $B \leftarrow$  Array of size  $r$   
4:   for  $w = f - r + 1$  to  $f$  do  
5:      $[z_w] = [c_w] - [d_{iw}]$   
6:      $B[w - f + r] = \text{INVERT-BIT}([z_w])$   
7:   end for  
8:    $[v_f] = \wedge_{h=1}^r B[h]$   
9:    $A[f - r + 1] = \text{INVERT-BIT}([v_f])$   
10: end for  
11: return  $\text{INVERT-BIT}(\wedge_{h=1}^{l-r+1} A[h])$ 
```

Algorithm 3 TOLERIZE($[D]$)

```
1: On receiving connection  $c$   
2:   Share  $c$  in bitwise additive fashion  
3: for  $i = 1$  to  $q$  do  
4:    $[u_i] = R - \text{CONT}([c], [d_i])$   
5:    $u_i = \text{RECONSTRUCT}([u_i])$   
6:   if  $u_i$  then  
7:     remove  $([d_i])$   
8:   end if  
9: end for
```
