Initial state
Game 1 is
(
  foreach $iH_{13} \leq qH$ **do**
  $OH(x : bitstring) :=$
  return($\mathsf{hash}(x)$)
|
  $Ogen() :=$
  $r \xleftarrow{R} seed$;
  $pk : pkey \leftarrow \mathsf{pkgen}(r)$;
  $sk : skey \leftarrow \mathsf{skgen}(r)$;
  return(pk);
  (
    foreach $iS_{14} \leq qS$ **do**
    $OS(m : bitstring) :=$
    return($\mathsf{invf}(sk, \mathsf{hash}(m))$)
  |
    $OT(m' : bitstring, s : D) :=$
    if ($\mathsf{f}(pk, s) = \mathsf{hash}(m')$) **then**
    find $u \leq qS$ **suchthat defined**($m[u]$) $\wedge$ ($m' = m[u]$) **then**
      end
    else
      event forge
)
)

Applying equivalence
foreach $iH_{11} \leq nH$ **do** $OH(x : bitstring) :=$ **return**($\mathsf{hash}(x)$)[all]
$\approx_0$
foreach $iH_{12} \leq nH$ **do** $OH(x_{16} : bitstring) := x : bitstring \leftarrow x_{16}$;
find $u \leq nH$ **suchthat defined**($x[u], r[u]$) $\wedge$ ($x = x[u]$) **then return**($r[u]$)
else $r \xleftarrow{R} D$; **return**(r)
yields

Game 2 is
(
  foreach $iH_{13} \leq qH$ **do**
  $OH(x : bitstring) :=$
  $x_{23} : bitstring \leftarrow x$;
  find **suchthat defined**(x_{19}, r_{18}) $\wedge$ ($x_{23} = x_{19}$) **then**
    return(r_{18})
  $\oplus$ $@i_{29} \leq qS$ **suchthat defined**($x_{21}[@i_{29}], r_{20}[@i_{29}]$) $\wedge$ ($x_{23} = x_{21}[@i_{29}]$) **then**
    return($r_{20}[@i_{29}]$)
  $\oplus$ $@i_{28} \leq qH$ **suchthat defined**($x_{23}[@i_{28}], r_{22}[@i_{28}]$) $\wedge$ ($x_{23} = x_{23}[@i_{28}]$) **then**
    return($r_{22}[@i_{28}]$)
  else
    $r_{22} \xleftarrow{R} D$;
    return(r_{22})
|
  $Ogen() :=$
  $r \xleftarrow{R} seed$;
  $pk : pkey \leftarrow \mathsf{pkgen}(r)$;

$sk : skey \leftarrow \mathsf{skgen}(r)$;
return(pk);
(
foreach $iS_{14} \leq qS$ **do**
$OS(m : bitstring) :=$
$x_{21} : bitstring \leftarrow m$;
find **suchthat** **defined**$(x_{19}, r_{18}) \wedge (x_{21} = x_{19})$ **then**
return$(\mathsf{invf}(sk, r_{18}))$
$\oplus$ $@i_{27} \leq qS$ **suchthat** **defined**$(x_{21}[@i_{27}], r_{20}[@i_{27}]) \wedge (x_{21} = x_{21}[@i_{27}])$ **then**
return$(\mathsf{invf}(sk, r_{20}[@i_{27}]))$
$\oplus$ $@i_{26} \leq qH$ **suchthat** **defined**$(x_{23}[@i_{26}], r_{22}[@i_{26}]) \wedge (x_{21} = x_{23}[@i_{26}])$ **then**
return$(\mathsf{invf}(sk, r_{22}[@i_{26}]))$
else
$r_{20} \stackrel{R}{\leftarrow} D$;
return$(\mathsf{invf}(sk, r_{20}))$
|
$OT(m' : bitstring, s : D) :=$
$x_{19} : bitstring \leftarrow m'$;
find **suchthat** **defined**$(x_{19}, r_{18}) \wedge (x_{19} = x_{19})$ **then**
if $(\mathsf{f}(pk, s) = r_{18})$ **then**
find $u \leq qS$ **suchthat** **defined**$(m[u]) \wedge (m' = m[u])$ **then**
end
else
event forge
$\oplus$ $@i_{25} \leq qS$ **suchthat** **defined**$(x_{21}[@i_{25}], r_{20}[@i_{25}]) \wedge (x_{19} = x_{21}[@i_{25}])$ **then**
if $(\mathsf{f}(pk, s) = r_{20}[@i_{25}])$ **then**
find $u \leq qS$ **suchthat** **defined**$(m[u]) \wedge (m' = m[u])$ **then**
end
else
event forge
$\oplus$ $@i_{24} \leq qH$ **suchthat** **defined**$(x_{23}[@i_{24}], r_{22}[@i_{24}]) \wedge (x_{19} = x_{23}[@i_{24}])$ **then**
if $(\mathsf{f}(pk, s) = r_{22}[@i_{24}])$ **then**
find $u \leq qS$ **suchthat** **defined**$(m[u]) \wedge (m' = m[u])$ **then**
end
else
event forge
else
$r_{18} \stackrel{R}{\leftarrow} D$;
if $(\mathsf{f}(pk, s) = r_{18})$ **then**
find $u \leq qS$ **suchthat** **defined**$(m[u]) \wedge (m' = m[u])$ **then**
end
else
event forge
)
)

Applying simplify yields

Game 3 is
(
foreach $iH_{13} \leq qH$ **do**
$OH(x : bitstring) :=$

```
  x_23 : bitstring ← x;
  find  suchthat defined(x_19, r_18) ∧ (x_23 = x_19) then
    return(r_18)
  ⊕ @i_29 ≤ qS suchthat defined(x_21[@i_29], r_20[@i_29]) ∧ (x_23 = x_21[@i_29]) then
    return(r_20[@i_29])
  ⊕ @i_28 ≤ qH suchthat defined(x_23[@i_28], r_22[@i_28]) ∧ (x_23 = x_23[@i_28]) then
    return(r_22[@i_28])
  else
    r_22 ←R D;
    return(r_22)
|
  Ogen() :=
  r ←R seed;
  pk : pkey ← pkgen(r);
  sk : skey ← skgen(r);
  return(pk);
  (
    foreach iS_14 ≤ qS do
    OS(m : bitstring) :=
    x_21 : bitstring ← m;
    find  suchthat defined(x_19, r_18) ∧ (x_21 = x_19) then
      return(invf(sk, r_18))
    ⊕ @i_27 ≤ qS suchthat defined(x_21[@i_27], r_20[@i_27]) ∧ (x_21 = x_21[@i_27]) then
      return(invf(sk, r_20[@i_27]))
    ⊕ @i_26 ≤ qH suchthat defined(x_23[@i_26], r_22[@i_26]) ∧ (x_21 = x_23[@i_26]) then
      return(invf(sk, r_22[@i_26]))
    else
      r_20 ←R D;
      return(invf(sk, r_20))
  |
    OT(m' : bitstring, s : D) :=
    x_19 : bitstring ← m';
    find @i_25 ≤ qS suchthat defined(r_20[@i_25], x_21[@i_25]) ∧ (x_19 = x_21[@i_25]) then
      end
    ⊕ @i_24 ≤ qH suchthat defined(x_23[@i_24], r_22[@i_24]) ∧ (x_19 = x_23[@i_24]) then
      if (f(pk, s) = r_22[@i_24]) then
      find u ≤ qS suchthat defined(m[u]) ∧ (m' = m[u]) then
        end
      else
        event forge
    else
      r_18 ←R D;
      if (f(pk, s) = r_18) then
      find u ≤ qS suchthat defined(m[u]) ∧ (m' = m[u]) then
        end
      else
        event forge
  )
)
```

Applying remove assignments of useless yields

Game 4 is
(
  foreach $iH_{13} \leq qH$ **do**
  $OH(x : bitstring) :=$
  $x_{23} : bitstring \leftarrow$ cst_bitstring;
  find suchthat defined$(m', x_{19}, r_{18}) \wedge (x = m')$ **then**
    return(r_{18})
  $\oplus$ $@i_{29} \leq qS$ **suchthat defined**$(m[@i_{29}], x_{21}[@i_{29}], r_{20}[@i_{29}]) \wedge (x = m[@i_{29}])$ **then**
    return$(r_{20}[@i_{29}])$
  $\oplus$ $@i_{28} \leq qH$ **suchthat defined**$(x[@i_{28}], x_{23}[@i_{28}], r_{22}[@i_{28}]) \wedge (x = x[@i_{28}])$ **then**
    return$(r_{22}[@i_{28}])$
  else
    $r_{22} \stackrel{R}{\leftarrow} D$;
    return(r_{22})
|
  $Ogen() :=$
  $r \stackrel{R}{\leftarrow} seed$;
  $pk : pkey \leftarrow \mathsf{pkgen}(r)$;
  $sk : skey \leftarrow \mathsf{skgen}(r)$;
  return(pk);
  (
    foreach $iS_{14} \leq qS$ **do**
    $OS(m : bitstring) :=$
    $x_{21} : bitstring \leftarrow$ cst_bitstring;
    find suchthat defined$(m', x_{19}, r_{18}) \wedge (m = m')$ **then**
      return$(\mathsf{invf}(sk, r_{18}))$
    $\oplus$ $@i_{27} \leq qS$ **suchthat defined**$(m[@i_{27}], x_{21}[@i_{27}], r_{20}[@i_{27}]) \wedge (m = m[@i_{27}])$ **then**
      return$(\mathsf{invf}(sk, r_{20}[@i_{27}]))$
    $\oplus$ $@i_{26} \leq qH$ **suchthat defined**$(x[@i_{26}], x_{23}[@i_{26}], r_{22}[@i_{26}]) \wedge (m = x[@i_{26}])$ **then**
      return$(\mathsf{invf}(sk, r_{22}[@i_{26}]))$
    else
      $r_{20} \stackrel{R}{\leftarrow} D$;
      return$(\mathsf{invf}(sk, r_{20}))$
  |
    $OT(m' : bitstring, s : D) :=$
    $x_{19} : bitstring \leftarrow$ cst_bitstring;
    find $@i_{25} \leq qS$ **suchthat defined**$(m[@i_{25}], x_{21}[@i_{25}], r_{20}[@i_{25}]) \wedge (m' = m[@i_{25}])$ **then**
      end
    $\oplus$ $@i_{24} \leq qH$ **suchthat defined**$(x[@i_{24}], x_{23}[@i_{24}], r_{22}[@i_{24}]) \wedge (m' = x[@i_{24}])$ **then**
      if $(\mathsf{f}(pk, s) = r_{22}[@i_{24}])$ **then**
      find $u \leq qS$ **suchthat defined**$(m[u]) \wedge (m' = m[u])$ **then**
        end
      else
        event forge
    else
      $r_{18} \stackrel{R}{\leftarrow} D$;
      if $(\mathsf{f}(pk, s) = r_{18})$ **then**
      find $u \leq qS$ **suchthat defined**$(m[u]) \wedge (m' = m[u])$ **then**
        end
      else
        event forge
)
)

Applying remove assignments of binder sk yields

Game 5 is
(
  foreach $iH_{13} \leq qH$ **do**
  $OH(x : bitstring) :=$
  $x_{23} : bitstring \leftarrow \mathsf{cst\_bitstring}$;
  find **suchthat** **defined**$(m', x_{19}, r_{18}) \wedge (x = m')$ **then**
    return(r_{18})
  $\oplus$ $@i_{29} \leq qS$ **suchthat** **defined**$(m[@i_{29}], x_{21}[@i_{29}], r_{20}[@i_{29}]) \wedge (x = m[@i_{29}])$ **then**
    return$(r_{20}[@i_{29}])$
  $\oplus$ $@i_{28} \leq qH$ **suchthat** **defined**$(x[@i_{28}], x_{23}[@i_{28}], r_{22}[@i_{28}]) \wedge (x = x[@i_{28}])$ **then**
    return$(r_{22}[@i_{28}])$
  else
    $r_{22} \xleftarrow{R} D$;
    return(r_{22})
|
  $Ogen() :=$
  $r \xleftarrow{R} seed$;
  $pk : pkey \leftarrow \mathsf{pkgen}(r)$;
  return(pk);
  (
    foreach $iS_{14} \leq qS$ **do**
    $OS(m : bitstring) :=$
    $x_{21} : bitstring \leftarrow \mathsf{cst\_bitstring}$;
    find **suchthat** **defined**$(m', x_{19}, r_{18}) \wedge (m = m')$ **then**
      return$(\mathsf{invf}(\mathsf{skgen}(r), r_{18}))$
    $\oplus$ $@i_{27} \leq qS$ **suchthat** **defined**$(m[@i_{27}], x_{21}[@i_{27}], r_{20}[@i_{27}]) \wedge (m = m[@i_{27}])$ **then**
      return$(\mathsf{invf}(\mathsf{skgen}(r), r_{20}[@i_{27}]))$
    $\oplus$ $@i_{26} \leq qH$ **suchthat** **defined**$(x[@i_{26}], x_{23}[@i_{26}], r_{22}[@i_{26}]) \wedge (m = x[@i_{26}])$ **then**
      return$(\mathsf{invf}(\mathsf{skgen}(r), r_{22}[@i_{26}]))$
    else
      $r_{20} \xleftarrow{R} D$;
      return$(\mathsf{invf}(\mathsf{skgen}(r), r_{20}))$
  |
    $OT(m' : bitstring, s : D) :=$
    $x_{19} : bitstring \leftarrow \mathsf{cst\_bitstring}$;
    find $@i_{25} \leq qS$ **suchthat** **defined**$(m[@i_{25}], x_{21}[@i_{25}], r_{20}[@i_{25}]) \wedge (m' = m[@i_{25}])$ **then**
      end
    $\oplus$ $@i_{24} \leq qH$ **suchthat** **defined**$(x[@i_{24}], x_{23}[@i_{24}], r_{22}[@i_{24}]) \wedge (m' = x[@i_{24}])$ **then**
      if $(\mathsf{f}(pk, s) = r_{22}[@i_{24}])$ **then**
      find $u \leq qS$ **suchthat** **defined**$(m[u]) \wedge (m' = m[u])$ **then**
        end
      else
        event forge
    else
      $r_{18} \xleftarrow{R} D$;
      if $(\mathsf{f}(pk, s) = r_{18})$ **then**
      find $u \leq qS$ **suchthat** **defined**$(m[u]) \wedge (m' = m[u])$ **then**
        end
      else

event forge
)
)

Applying equivalence
foreach $iK_1 \leq nK$ **do** $r \xleftarrow{R} seed$; (
$Opk() :=$ **return**(pkgen(r)) |
foreach $iF_2 \leq nF$ **do** $x \xleftarrow{R} D$; (
$Oant() :=$ **return**(invf(skgen(r), x)) |
$Oim() :=$ **return**(x)))
$\approx_0$
foreach $iK_3 \leq nK$ **do** $r \xleftarrow{R} seed$; (
$Opk() :=$ **return**(pkgen(r)) |
foreach $iF_4 \leq nF$ **do** $x \xleftarrow{R} D$; (
$Oant() :=$ **return**(x) |
$Oim() :=$ **return**(f(pkgen(r), x))))
with r yields

Game 6 is
(
foreach $iH_{13} \leq qH$ **do**
$OH(x : bitstring) :=$
$x_{23} : bitstring \leftarrow$ cst_bitstring;
find suchthat defined$(m', x_{19}, r_{18}) \wedge (x = m')$ **then**
return(f(pkgen(r), r_{18}))
$\oplus$ $@i_{29} \leq qS$ **suchthat defined**$(m[@i_{29}], x_{21}[@i_{29}], r_{20}[@i_{29}]) \wedge (x = m[@i_{29}])$ **then**
return(f(pkgen(r), $r_{20}[@i_{29}]$))
$\oplus$ $@i_{28} \leq qH$ **suchthat defined**$(x[@i_{28}], x_{23}[@i_{28}], r_{22}[@i_{28}]) \wedge (x = x[@i_{28}])$ **then**
return(f(pkgen(r), $r_{22}[@i_{28}]$))
else
$r_{22} \xleftarrow{R} D$;
return(f(pkgen(r), r_{22}))
|
$Ogen() :=$
$r \xleftarrow{R} seed$;
$pk : pkey \leftarrow$ pkgen(r);
return(pk);
(
foreach $iS_{14} \leq qS$ **do**
$OS(m : bitstring) :=$
$x_{21} : bitstring \leftarrow$ cst_bitstring;
find suchthat defined$(m', x_{19}, r_{18}) \wedge (m = m')$ **then**
return(r_{18})
$\oplus$ $@i_{27} \leq qS$ **suchthat defined**$(m[@i_{27}], x_{21}[@i_{27}], r_{20}[@i_{27}]) \wedge (m = m[@i_{27}])$ **then**
return($r_{20}[@i_{27}]$)
$\oplus$ $@i_{26} \leq qH$ **suchthat defined**$(x[@i_{26}], x_{23}[@i_{26}], r_{22}[@i_{26}]) \wedge (m = x[@i_{26}])$ **then**
return($r_{22}[@i_{26}]$)
else
$r_{20} \xleftarrow{R} D$;
return(r_{20})
|

$OT(m' : bitstring, s : D) :=$
$x_{19} : bitstring \leftarrow$ cst_bitstring;
find $@i_{25} \leq qS$ **suchthat defined**$(m[@i_{25}], x_{21}[@i_{25}], r_{20}[@i_{25}]) \wedge (m' = m[@i_{25}])$ **then**
end
$\oplus$ $@i_{24} \leq qH$ **suchthat defined**$(x[@i_{24}], x_{23}[@i_{24}], r_{22}[@i_{24}]) \wedge (m' = x[@i_{24}])$ **then**
if $(\mathsf{f}(pk, s) = \mathsf{f}(\mathsf{pkgen}(r), r_{22}[@i_{24}]))$ **then**
find $u \leq qS$ **suchthat defined**$(m[u]) \wedge (m' = m[u])$ **then**
end
else
event forge
else
$r_{18} \stackrel{R}{\leftarrow} D$;
if $(\mathsf{f}(pk, s) = \mathsf{f}(\mathsf{pkgen}(r), r_{18}))$ **then**
find $u \leq qS$ **suchthat defined**$(m[u]) \wedge (m' = m[u])$ **then**
end
else
event forge
)
)

Applying simplify yields

Game 7 is
(
foreach $iH_{13} \leq qH$ **do**
$OH(x : bitstring) :=$
$x_{23} : bitstring \leftarrow$ cst_bitstring;
find **suchthat defined**$(m', r, r_{18}) \wedge (x = m')$ **then**
return$(\mathsf{f}(\mathsf{pkgen}(r), r_{18}))$
$\oplus$ $@i_{29} \leq qS$ **suchthat defined**$(m[@i_{29}], r, r_{20}[@i_{29}]) \wedge (x = m[@i_{29}])$ **then**
return$(\mathsf{f}(\mathsf{pkgen}(r), r_{20}[@i_{29}]))$
$\oplus$ $@i_{28} \leq qH$ **suchthat defined**$(x[@i_{28}], r_{22}[@i_{28}]) \wedge (x = x[@i_{28}])$ **then**
return$(\mathsf{f}(\mathsf{pkgen}(r), r_{22}[@i_{28}]))$
else
$r_{22} \stackrel{R}{\leftarrow} D$;
return$(\mathsf{f}(\mathsf{pkgen}(r), r_{22}))$
|
$Ogen() :=$
$r \stackrel{R}{\leftarrow} seed$;
$pk : pkey \leftarrow \mathsf{pkgen}(r)$;
return(pk);
(
foreach $iS_{14} \leq qS$ **do**
$OS(m : bitstring) :=$
$x_{21} : bitstring \leftarrow$ cst_bitstring;
find **suchthat defined**$(m', r_{18}) \wedge (m = m')$ **then**
return(r_{18})
$\oplus$ $@i_{27} \leq qS$ **suchthat defined**$(m[@i_{27}], r_{20}[@i_{27}]) \wedge (m = m[@i_{27}])$ **then**
return$(r_{20}[@i_{27}])$
$\oplus$ $@i_{26} \leq qH$ **suchthat defined**$(x[@i_{26}], r_{22}[@i_{26}]) \wedge (m = x[@i_{26}])$ **then**
return$(r_{22}[@i_{26}])$
else

$r_{20} \stackrel{R}{\leftarrow} D$;
return(r_{20})
|
$OT(m' : bitstring, s : D) :=$
$x_{19} : bitstring \leftarrow$ cst_bitstring;
find $@i_{25} \leq qS$ **suchthat defined**($r_{20}[@i_{25}], m[@i_{25}]) \wedge (m' = m[@i_{25}])$ **then**
end
$\oplus$ $@i_{24} \leq qH$ **suchthat defined**($x[@i_{24}], r_{22}[@i_{24}]) \wedge (m' = x[@i_{24}])$ **then**
if $(s = r_{22}[@i_{24}])$ **then**
find $u \leq qS$ **suchthat defined**($m[u]) \wedge (m' = m[u])$ **then**
end
else
event forge
else
$r_{18} \stackrel{R}{\leftarrow} D$;
if $(s = r_{18})$ **then**
find $u \leq qS$ **suchthat defined**($m[u]) \wedge (m' = m[u])$ **then**
end
else
event forge
)
)

Applying remove assignments of useless yields

Game 8 is
(
foreach $iH_{13} \leq qH$ **do**
$OH(x : bitstring) :=$
find suchthat defined($m', r, r_{18}) \wedge (x = m')$ **then**
return(f(pkgen(r), r_{18}))
$\oplus$ $@i_{29} \leq qS$ **suchthat defined**($m[@i_{29}], r, r_{20}[@i_{29}]) \wedge (x = m[@i_{29}])$ **then**
return(f(pkgen(r), $r_{20}[@i_{29}]$))
$\oplus$ $@i_{28} \leq qH$ **suchthat defined**($x[@i_{28}], r_{22}[@i_{28}]) \wedge (x = x[@i_{28}])$ **then**
return(f(pkgen(r), $r_{22}[@i_{28}]$))
else
$r_{22} \stackrel{R}{\leftarrow} D$;
return(f(pkgen(r), r_{22}))
|
$Ogen() :=$
$r \stackrel{R}{\leftarrow} seed$;
$pk : pkey \leftarrow$ pkgen(r);
return(pk);
(
foreach $iS_{14} \leq qS$ **do**
$OS(m : bitstring) :=$
find suchthat defined($m', r_{18}) \wedge (m = m')$ **then**
return(r_{18})
$\oplus$ $@i_{27} \leq qS$ **suchthat defined**($m[@i_{27}], r_{20}[@i_{27}]) \wedge (m = m[@i_{27}])$ **then**
return($r_{20}[@i_{27}]$)
$\oplus$ $@i_{26} \leq qH$ **suchthat defined**($x[@i_{26}], r_{22}[@i_{26}]) \wedge (m = x[@i_{26}])$ **then**
return($r_{22}[@i_{26}]$)

else
$r_{20} \stackrel{R}{\leftarrow} D$;
return(r_{20})
|
$OT(m' : bitstring, s : D) :=$
find $@i_{25} \leq qS$ **suchthat defined**$(r_{20}[@i_{25}], m[@i_{25}]) \wedge (m' = m[@i_{25}])$ **then**
end
$\oplus$ $@i_{24} \leq qH$ **suchthat defined**$(x[@i_{24}], r_{22}[@i_{24}]) \wedge (m' = x[@i_{24}])$ **then**
if $(s = r_{22}[@i_{24}])$ **then**
find $u \leq qS$ **suchthat defined**$(m[u]) \wedge (m' = m[u])$ **then**
end
else
event forge
else
$r_{18} \stackrel{R}{\leftarrow} D$;
if $(s = r_{18})$ **then**
find $u \leq qS$ **suchthat defined**$(m[u]) \wedge (m' = m[u])$ **then**
end
else
event forge
)
)

Applying equivalence
foreach $iK_5 \leq nK$ **do** $r \stackrel{R}{\leftarrow} seed$; (
$Opk() :=$ **return**(pkgen(r)) |
foreach $iF_6 \leq nF$ **do** $x \stackrel{R}{\leftarrow} D$; (
$Oy() :=$ **return**(f(pkgen$(r), x)$) |
foreach $i1_7 \leq n1$ **do** $Oeq(x' : D) :=$ **return**$((x' = x))$ |
$Ox() :=$ **return**(x))))
$\approx_{nK \times nF \times POW(\mathbf{time} + (nK-1.) \times \mathbf{time}(\mathsf{pkgen}) + (nF \times nK - 1.) \times \mathbf{time}(\mathsf{f}))}$
foreach $iK_8 \leq nK$ **do** $r \stackrel{R}{\leftarrow} seed$; (
$Opk() :=$ **return**(pkgen$'(r)$) |
foreach $iF_9 \leq nF$ **do** $x \stackrel{R}{\leftarrow} D$; (
$Oy() :=$ **return**(f$'$(pkgen$'(r), x)$) |
foreach $i1_{10} \leq n1$ **do** $Oeq(x' : D) :=$ **if defined**(k) **then return**$((x' = x))$ **else return**(false) |
$Ox() := k : bitstring \leftarrow$ mark; **return**(x)))
[Difference of probability $POW(qS \times \mathbf{time}(\mathsf{f}) + qH \times \mathbf{time}(\mathsf{f}) + \mathbf{time} + \mathbf{time}(\textit{context for game } 8)) +$
$qS \times POW(qS \times \mathbf{time}(\mathsf{f}) + qH \times \mathbf{time}(\mathsf{f}) + \mathbf{time} + \mathbf{time}(\textit{context for game } 8)) +$
$qH \times POW(qS \times \mathbf{time}(\mathsf{f}) + qH \times \mathbf{time}(\mathsf{f}) + \mathbf{time} + \mathbf{time}(\textit{context for game } 8))$] yields

Game 9 is
(
foreach $iH_{13} \leq qH$ **do**
$OH(x : bitstring) :=$
find **suchthat defined**$(m', r, r_{18}) \wedge (x = m')$ **then**
return(f$'$(pkgen$'(r), r_{18})$)
$\oplus$ $@i_{29} \leq qS$ **suchthat defined**$(m[@i_{29}], r, r_{20}[@i_{29}]) \wedge (x = m[@i_{29}])$ **then**
return(f$'$(pkgen$'(r), r_{20}[@i_{29}])$)
$\oplus$ $@i_{28} \leq qH$ **suchthat defined**$(x[@i_{28}], r_{22}[@i_{28}]) \wedge (x = x[@i_{28}])$ **then**
return(f$'$(pkgen$'(r), r_{22}[@i_{28}])$)

```
else
  r22 <-R- D;
  return(f'(pkgen'(r), r22))
|
Ogen() :=
r <-R- seed;
pk : pkey <- pkgen'(r);
return(pk);
(
  foreach iS14 ≤ qS do
  OS(m : bitstring) :=
  find  suchthat defined(m', r18) ∧ (m = m') then
    k47 : bitstring <- mark;
    return(r18)
  ⊕ @i27 ≤ qS suchthat defined(m[@i27], r20[@i27]) ∧ (m = m[@i27]) then
    k48 : bitstring <- mark;
    return(r20[@i27])
  ⊕ @i26 ≤ qH suchthat defined(x[@i26], r22[@i26]) ∧ (m = x[@i26]) then
    k50 : bitstring <- mark;
    return(r22[@i26])
  else
    r20 <-R- D;
    k45 : bitstring <- mark;
    return(r20)
|
  OT(m' : bitstring, s : D) :=
  find @i25 ≤ qS suchthat defined(r20[@i25], m[@i25]) ∧ (m' = m[@i25]) then
    end
  ⊕ @i24 ≤ qH suchthat defined(x[@i24], r22[@i24]) ∧ (m' = x[@i24]) then
    find @i56 ≤ qS suchthat defined(k50[@i56]) ∧ (@i24 = @i26[@i56]) then
      if (s = r22[@i24]) then
      find u ≤ qS suchthat defined(m[u]) ∧ (m' = m[u]) then
        end
      else
        event forge
    else
      if false then
      find u ≤ qS suchthat defined(m[u]) ∧ (m' = m[u]) then
        end
      else
        event forge
  else
    r18 <-R- D;
    find @i53 ≤ qS suchthat defined(k47[@i53]) then
      if (s = r18) then
      find u ≤ qS suchthat defined(m[u]) ∧ (m' = m[u]) then
        end
      else
        event forge
    else
      if false then
      find u ≤ qS suchthat defined(m[u]) ∧ (m' = m[u]) then
        end
```

```
        else
          event forge
  )
)
```

Applying simplify yields

Game 10 is

```
(
  foreach iH_13 ≤ qH do
  OH(x : bitstring) :=
  find  suchthat defined(m', r, r_18) ∧ (x = m') then
    return(f'(pkgen'(r), r_18))
  ⊕ @i_29 ≤ qS suchthat defined(m[@i_29], r, r_20[@i_29]) ∧ (x = m[@i_29]) then
    return(f'(pkgen'(r), r_20[@i_29]))
  ⊕ @i_28 ≤ qH suchthat defined(x[@i_28], r_22[@i_28]) ∧ (x = x[@i_28]) then
    return(f'(pkgen'(r), r_22[@i_28]))
  else
    r_22 ←R D;
    return(f'(pkgen'(r), r_22))
|
  Ogen() :=
  r ←R seed;
  pk : pkey ← pkgen'(r);
  return(pk);
  (
    foreach iS_14 ≤ qS do
    OS(m : bitstring) :=
    find  suchthat defined(m', r_18) ∧ (m = m') then
      k_47 : bitstring ← mark;
      return(r_18)
    ⊕ @i_27 ≤ qS suchthat defined(m[@i_27], r_20[@i_27]) ∧ (m = m[@i_27]) then
      k_48 : bitstring ← mark;
      return(r_20[@i_27])
    ⊕ @i_26 ≤ qH suchthat defined(x[@i_26], r_22[@i_26]) ∧ (m = x[@i_26]) then
      k_50 : bitstring ← mark;
      return(r_22[@i_26])
    else
      r_20 ←R D;
      k_45 : bitstring ← mark;
      return(r_20)
  |
    OT(m' : bitstring, s : D) :=
    find @i_25 ≤ qS suchthat defined(r_20[@i_25], m[@i_25]) ∧ (m' = m[@i_25]) then
      end
    ⊕ @i_24 ≤ qH suchthat defined(r_22[@i_24], x[@i_24]) ∧ (m' = x[@i_24]) then
      end
    else
      r_18 ←R D
  )
)
```

Applying remove assignments of useless yields

Game 11 is
(
  foreach $iH_{13} \leq qH$ **do**
  $OH(x : bitstring) :=$
  find **suchthat** **defined**$(m', r, r_{18}) \wedge (x = m')$ **then**
    return$(\mathsf{f}'(\mathsf{pkgen}'(r), r_{18}))$
  $\oplus$ $@i_{29} \leq qS$ **suchthat** **defined**$(m[@i_{29}], r, r_{20}[@i_{29}]) \wedge (x = m[@i_{29}])$ **then**
    return$(\mathsf{f}'(\mathsf{pkgen}'(r), r_{20}[@i_{29}]))$
  $\oplus$ $@i_{28} \leq qH$ **suchthat** **defined**$(x[@i_{28}], r_{22}[@i_{28}]) \wedge (x = x[@i_{28}])$ **then**
    return$(\mathsf{f}'(\mathsf{pkgen}'(r), r_{22}[@i_{28}]))$
  else
    $r_{22} \xleftarrow{R} D$;
    return$(\mathsf{f}'(\mathsf{pkgen}'(r), r_{22}))$
|
  $Ogen() :=$
  $r \xleftarrow{R} seed$;
  $pk : pkey \leftarrow \mathsf{pkgen}'(r)$;
  return(pk);
  (
    foreach $iS_{14} \leq qS$ **do**
    $OS(m : bitstring) :=$
    find **suchthat** **defined**$(m', r_{18}) \wedge (m = m')$ **then**
      return(r_{18})
    $\oplus$ $@i_{27} \leq qS$ **suchthat** **defined**$(m[@i_{27}], r_{20}[@i_{27}]) \wedge (m = m[@i_{27}])$ **then**
      return$(r_{20}[@i_{27}])$
    $\oplus$ $@i_{26} \leq qH$ **suchthat** **defined**$(x[@i_{26}], r_{22}[@i_{26}]) \wedge (m = x[@i_{26}])$ **then**
      return$(r_{22}[@i_{26}])$
    else
      $r_{20} \xleftarrow{R} D$;
      return(r_{20})
  |
    $OT(m' : bitstring, s : D) :=$
    find $@i_{25} \leq qS$ **suchthat** **defined**$(r_{20}[@i_{25}], m[@i_{25}]) \wedge (m' = m[@i_{25}])$ **then**
      end
    $\oplus$ $@i_{24} \leq qH$ **suchthat** **defined**$(r_{22}[@i_{24}], x[@i_{24}]) \wedge (m' = x[@i_{24}])$ **then**
      end
    else
      $r_{18} \xleftarrow{R} D$
)
)
Proved event $\mathsf{forge} \Longrightarrow \mathsf{false}$ with probability
$qH \times POW(qS \times \mathbf{time}(\mathsf{f}) + qH \times \mathbf{time}(\mathsf{f}) + \mathbf{time} + \mathbf{time}(\textit{context for game } 8))+$
$qS \times POW(qS \times \mathbf{time}(\mathsf{f}) + qH \times \mathbf{time}(\mathsf{f}) + \mathbf{time} + \mathbf{time}(\textit{context for game } 8))+$
$POW(qS \times \mathbf{time}(\mathsf{f}) + qH \times \mathbf{time}(\mathsf{f}) + \mathbf{time} + \mathbf{time}(\textit{context for game } 8))$
RESULT $\mathbf{time}(\textit{context for game } 8) =$
$2. \times qS \times \mathbf{time}(=\!bitstring, \mathbf{maxlength}(\textit{game } 8 : m'), \mathbf{maxlength}(\textit{game } 8 : m[iS_{14}]))+$
$qH \times \mathbf{time}(=\!bitstring, \mathbf{maxlength}(\textit{game } 8 : m'), \mathbf{maxlength}(\textit{game } 8 : x[iH_{13}]))+$
$qH \times \mathbf{time}(=\!bitstring, \mathbf{maxlength}(\textit{game } 8 : m[iS_{14}]), \mathbf{maxlength}(\textit{game } 8 : x[iH_{13}])) \times qS+$
$qS \times qS \times \mathbf{time}(=\!bitstring, \mathbf{maxlength}(\textit{game } 8 : m[iS_{14}]), \mathbf{maxlength}(\textit{game } 8 : m[iS_{14}]))+$
$\mathbf{time}(=\!bitstring, \mathbf{maxlength}(\textit{game } 8 : m[iS_{14}]), \mathbf{maxlength}(\textit{game } 8 : m')) \times qS+$

$qH \times qH \times \mathbf{time}(=\mathit{bitstring}, \mathbf{maxlength}(\mathit{game}\ 8 : x[iH_{13}]), \mathbf{maxlength}(\mathit{game}\ 8 : x[iH_{13}]))+$
$qS \times \mathbf{time}(=\mathit{bitstring}, \mathbf{maxlength}(\mathit{game}\ 8 : x[iH_{13}]), \mathbf{maxlength}(\mathit{game}\ 8 : m[iS_{14}])) \times qH+$
$\mathbf{time}(=\mathit{bitstring}, \mathbf{maxlength}(\mathit{game}\ 8 : x[iH_{13}]), \mathbf{maxlength}(\mathit{game}\ 8 : m')) \times qH$
All queries proved.