A trace has been found.

Abbreviations
~M = xor(xor(km,kw),r1_3)
~M_1 = xor(disjunction(ID_m,ID_w),r2_3)
~M_2 = concat(xor(kw,rot(xor(km,concat(r2_3,km)))), xor(km,rot(xor(kw,concat(r1_3,kw)))))
~M_3 = h(btAddr_3)

Honest Process   Attacker

{1} new btAddr_3
{2} new k_qr_3

!   !

Beginning of process WD
{4} let id_w: id = ID_w
{8} event WDStartsSession(ID_w,ID_m)
{9} new r1_4
{10} new r2_4
{12} let r2_bits: bitstring = r2_4
{11} let r1_bits: bitstring = r1_4
{7} let k_qr_1: secKey = k_qr_3
{5} let k_w: bitstring = kw
{13} let qr1: qrcode = qrenc((kw,r1_4,r2_4),k_qr_3)

Beginning of process WD
{4} let id_w: id = ID_w
{8} event WDStartsSession(ID_w,ID_m)
{9} new r1_3
{10} new r2_3
{12} let r2_bits: bitstring = r2_3
{11} let r1_bits: bitstring = r1_3
{7} let k_qr_1: secKey = k_qr_3
{5} let k_w: bitstring = kw
{13} let qr1: qrcode = qrenc((kw,r1_3,r2_3),k_qr_3)

Beginning of process MT
{33} let id_m: id = ID_m
{37} event MTStartsSession(ID_m,ID_w)

qrenc((kw,r1_3,r2_3),k_qr_3)

{36} let k_qr_2: secKey = k_qr_3
{39} let (k_w_1: bitstring,r1_bits_1: bitstring, r2_bits_1: bitstring) = (kw,r1_3,r2_3)
{34} let k_m: bitstring = km
{43} let k_m'_1: bitstring = rot(xor(km,concat(r2_3,km)))
{42} let k_w'_1: bitstring = rot(xor(kw,concat(r1_3,kw)))
{44} let C_1: bitstring = concat(xor(kw,rot(xor(km,concat(r2_3,km)))),xor(km,rot(xor(kw,concat(r1_3,kw)))))
{41} let B_1: bitstring = xor(disjunction(ID_m,ID_w),r2_3)
{40} let A_1: bitstring = xor(xor(km,kw),r1_3)
{35} let btAddr_2: bitstring = btAddr_3
{45} event MTSendsBluetooth(xor(xor(km,kw),r1_3),xor(disjunction(ID_m,ID_w),r2_3),concat(xor(kw,rot(xor(km,concat(r2_3,km)))),xor(km,rot(xor(kw,concat(r1_3,kw))))),h(btAddr_3))

(~M,~M_1,~M_2,~M_3)

qrenc((kw,r1_4,r2_4),k_qr_3)

{48} let M1_1: bitstring = (kw,r1_4,r2_4)

{49} event MTReceivesQR2(ID_w,ID_m,(kw,r1_4,r2_4))