Formal verification of Isawa-Morii authentication protocol*

Mehmet Tahir Sandıkkaya

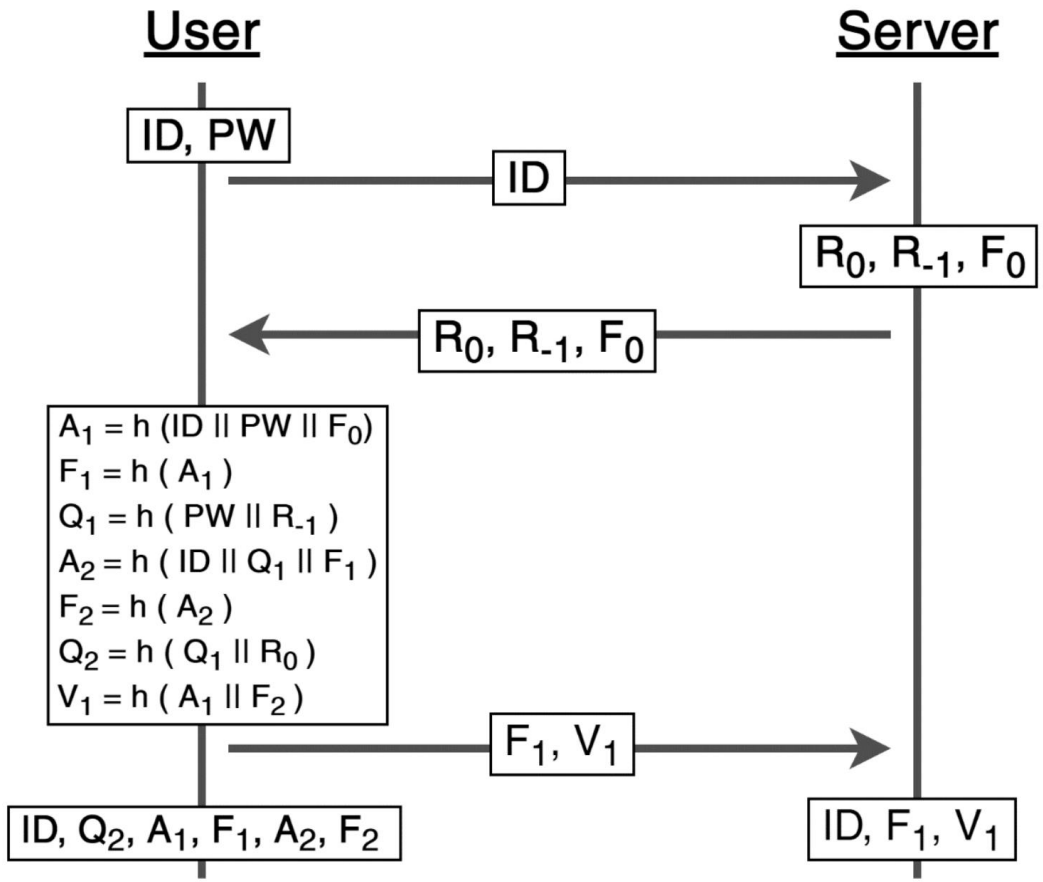
Computer Engineering Department
Istanbul Technical University

* *One-Time Password Authentication Scheme to Solve Stolen Verifier Problem* by Ryoichi Isawa and Masakatu Morii

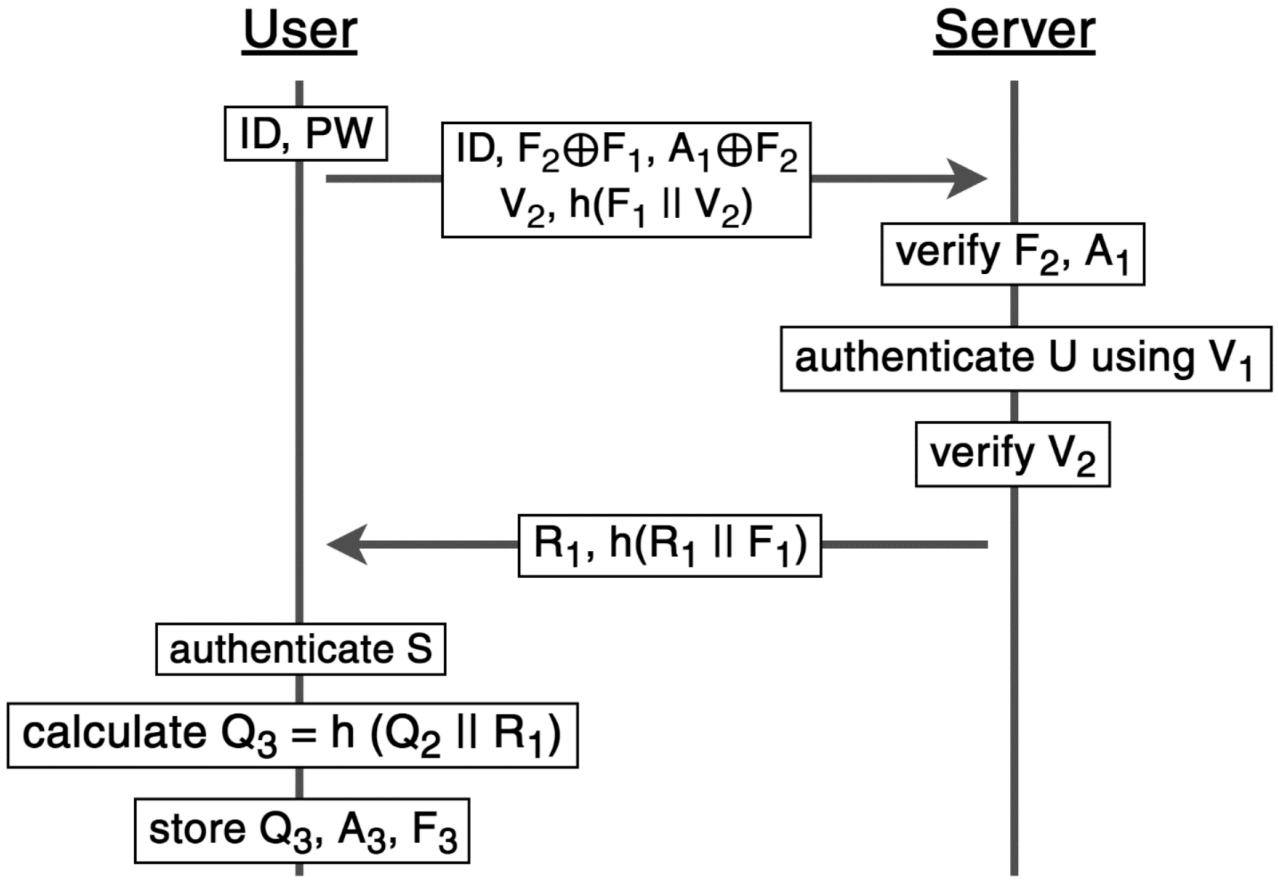
Basics of the protocol

- Very lightweight and interactive, could be used in IoT context
- hash function
  - one-way function
  - common for every OTP protocol
  - verification of the current protocol state in a future state
  - output is indistinguishable from random oracle
- XOR
  - replaces encryption
  - information-theoretically secure (!)
  - could mask values temporarily

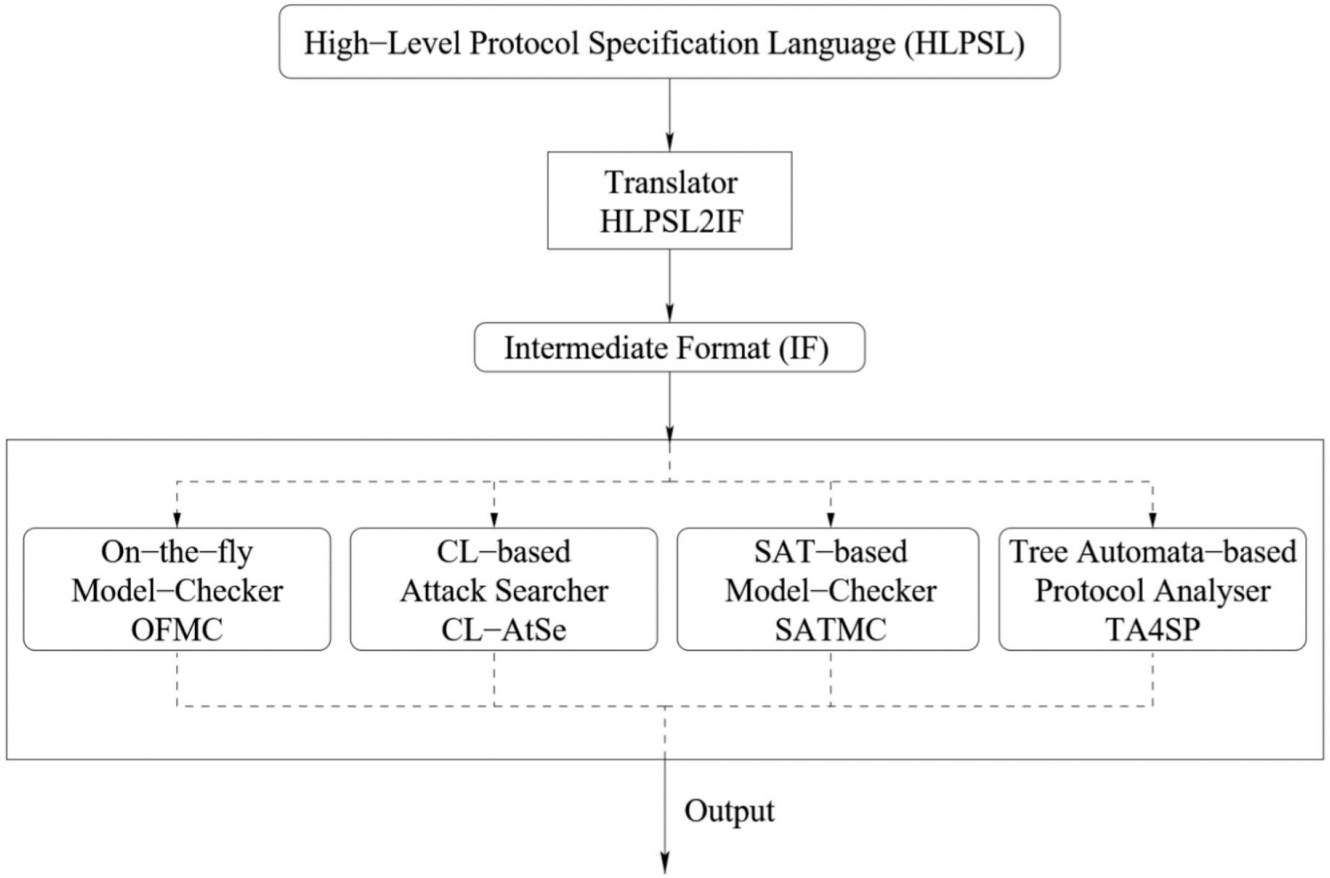
Registration



Authentication



Tool: AVISPA* (Automated Validation of Internet Security Protocols and Applications)



* http://www.avispa-project.org

User role parameters and values

```
role user(
    U,S             : agent,
    Kus             : symmetric_key,
    H               : hash_func,
    SND,RCV         : channel (dy))
played_by U
def=

    local
        State       : nat,
        ID          : text,
        PW          : text,
        A1          : hash(text.text.text),
        A2          : hash(text.hash(text.text).text),
        F1          : hash(hash(text.text.text)),
        F2          : hash(hash(text.hash(text.text).text)),
        Q2          : hash(hash(text.text).text),
        Q3          : hash(hash(hash(text.text).text).text),
        A3          : hash(text.hash(hash(text.text).text).hash(hash(text.hash(text.text).text))),
        F3          : hash(hash(text.hash(hash(text.text).text).hash(hash(text.hash(text.text).text)))),
        R0,RM,F0    : text,
        R1          : text
        S
    init
        State       := 0
```

User role states

```
transition
    1. State = 0     /\ RCV(start)
    =|>
    State' := 2      /\ ID' := new()
                     /\ PW' := new()
                     /\ SND({ID'}_Kus)
                     /\ secret(PW',sec_PW,{U})
                     /\ secret(ID',sec_ID,{U,S})

    2. State = 2     /\ RCV({R0'.RM'.F0'}_Kus)
    =|>
    State' := 4      /\ A1' := H(ID.PW.F0')
                     /\ F1' := H(A1')
                     /\ A2' := H(ID.H(PW.RM').F1')
                     /\ F2' := H(A2')
                     /\ Q2' := H(H(PW.RM').R0')
                     /\ SND({H(H(ID.PW.F0')).H(H(ID.PW.F0').H(H(ID.H(PW.RM').H(H(ID.PW.F0')))))}_Kus)

    3. State = 4
    =|>
    State' := 6      /\ A3' := H(ID.Q2.F2)
                     /\ F3' := H(A3')
                     /\ SND(ID)
```

User role states

```
4. State = 6
=|>
State' := 8 /\ SND(xor(H(H(ID.H(PW.RM).H(H(ID.PW.F0)))),H(H(ID.PW.F0))))

5. State = 8
=|>
State' := 10    /\ SND(xor(H(ID.PW.F0),H(H(ID.PW.F0).H(H(ID.H(PW.RM).H(H(ID.PW.F0)))))))

6. State = 10
=|>
State' := 12    /\ SND(H(H(H(ID.H(PW.RM).H(H(ID.PW.F0)))).H(H(ID.H(H(PW.RM).R0).H(H(ID.H(PW.RM).H(H(ID.PW.F0))))))))

7. State = 12
=|>
State' := 14    /\ SND(H(H(H(ID.PW.F0)).H(H(H(ID.H(PW.RM).H(H(ID.PW.F0)))).H(H(ID.H(H(PW.RM).R0).H(H(ID.H(PW.RM).H(H(ID.PW.F0))))))))))

8. State = 14   /\ RCV(R1')
=|>
State' := 16

9. State = 16   /\ RCV(H(R1.F1))
=|>
State' := 18    /\ Q3' := H(Q2.R1)
```

Server role parameters and values

```
role server(
    S,U             : agent,
    Kus             : symmetric_key,
    H               : hash_func,
    SND,RCV         : channel (dy))
played_by S
def=

    local
        State       : nat,
        ID          : text,
        R0,RM,F0    : text,
        PW          : text,
        F1          : hash(hash(text.text.text)),
        V1          : hash(hash(text.text.text).hash(hash(text.hash(text.text).hash(hash(text.text.text))))),
        F2          : hash(hash(text.hash(text.text).text)),
        A1          : hash(text.text.text),
        F1C         : hash(hash(text.text.text)),
        V2          : hash(hash(text.hash(text.text).text).hash(hash(text.hash(hash(text.text).text).hash(hash(text.hash(text.text).text
        V1C         : hash(hash(text.text.text).hash(hash(text.hash(text.text).hash(hash(text.text.text))))),
        R1          : text

    init
        State       := 1
```

Server role states

```
transition
    1. State = 1     /\ RCV({ID'}_Kus)
    =|>
    State' := 3      /\ R0' := new()
                     /\ RM' := new()
                     /\ F0' := new()
                     /\ SND({R0'.RM'.F0'}_Kus)
                     /\ secret(R0',sec_R0,{U,S})
                     /\ secret(RM',sec_RM,{U,S})
                     /\ secret(F0',sec_F0,{U,S})

    2. State = 3     /\ RCV({H(H(ID.PW'.F0)).H(H(ID.PW'.F0).H(H(ID.H(PW'.RM).H(H(ID.PW'.F0)))))}_Kus)
    =|>
    State' := 5      /\ F1' := H(H(ID.PW'.F0))
                     /\ V1' := H(H(ID.PW'.F0).H(H(ID.H(PW'.RM).H(H(ID.PW'.F0)))))

    3. State = 5     /\ RCV(ID)
    =|>
    State' := 7
```

Server role states

```
4. State = 7     /\ RCV(xor(F2',F1))
=|>
State' := 9

5. State = 9     /\ RCV(xor(A1',H(H(ID.PW.F0).H(H(ID.H(PW.RM).H(H(ID.PW.F0)))))))
=|>
State' := 11     /\ F1C' := H(A1')

6. State = 11    /\ RCV(H(H(H(ID.H(PW.RM).H(H(ID.PW.F0)))).H(H(ID.H(H(PW.RM).R0).H(H(ID.H(PW.RM).H(H(ID.PW.F0))))))))
                 /\ F1 = F1C
=|>
State' := 13     /\ V1C' := H(A1.F2)
                 /\ V2' := H(H(H(ID.H(PW.RM).H(H(ID.PW.F0)))).H(H(ID.H(H(PW.RM).R0).H(H(ID.H(PW.RM).H(H(ID.PW.F0)))))))

7. State = 13    /\ RCV(H(F1.V2))
                 /\ V1 = V1C
=|>
State' := 15     /\ R1' := new()
                 /\ SND(R1')

8. State = 15
=|>
State' := 17     /\ SND(H(R1.F1))
```

Session role

```
role session(
    U,S             : agent,
    Kus             : symmetric_key,
    H               : hash_func)
def=

    local
        RU,SU       : channel (dy),
        RS,SS       : channel (dy)

    composition
        user(U,S,Kus,H,SU,RU)
    /\  server(S,U,Kus,H,SS,RS)

end role
```

Environment role

```
role environment()
def=

    const
        u,s          : agent,
        kus          : symmetric_key,
        kis,kui      : symmetric_key,
        h            : hash_func,
        sec_R0       : protocol_id,
        sec_RM       : protocol_id,
        sec_F0       : protocol_id,
        sec_PW       : protocol_id,
        sec_ID       : protocol_id

    intruder_knowledge={u,s,h,kis,kui}

    composition
        session(u,s,kus,h)
    /\  session(u,s,kus,h)
    /\  session(u,i,kui,h)
    /\  session(i,s,kis,h)


end role
```

Goals

```
goal
    secrecy_of sec_PW
%   secrecy_of sec_ID
    secrecy_of sec_R0
    secrecy_of sec_RM
    secrecy_of sec_F0
end goal
```