A novel approach for securing data against intrusion attacks in unmanned aerial vehicles integrated heterogeneous network using functional encryption technique

Transaction on Emerging Telecommunications Technologies (ETT) '20

Diwankshi Sharma, Sachin Kumar Gupta, Aabid Rashid,
Sumeet Gupta, Mamoon Rashid, Ashutosh Srivastava

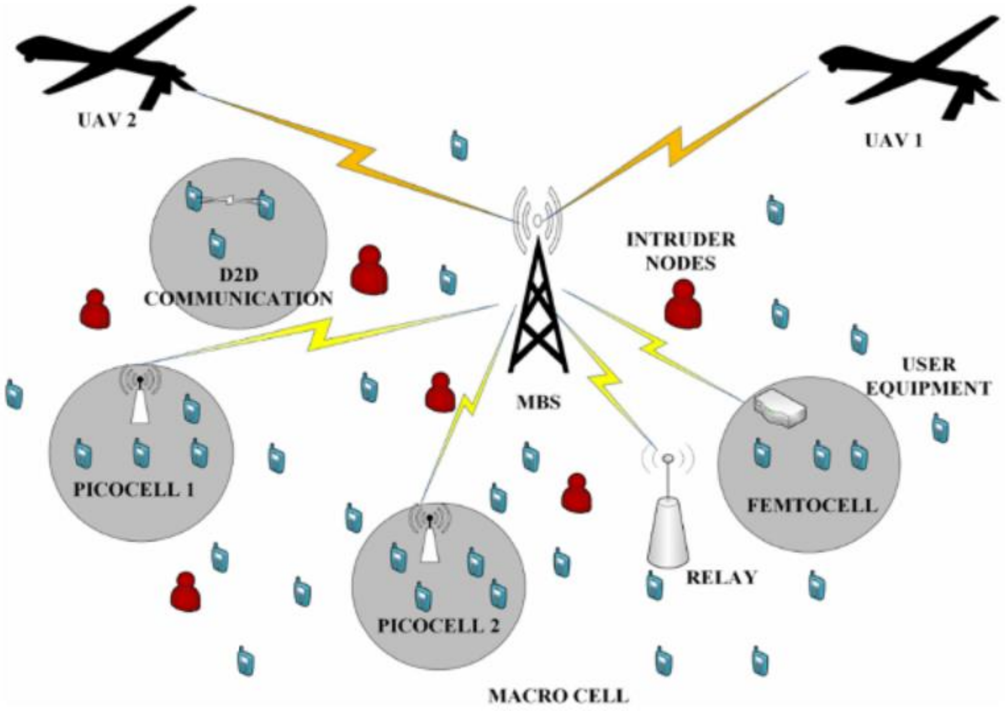
JaeHyun Lee (jhlee2021@mmlab.snu.ac.kr)

Content

- Motivation
- Background (Functional Encryption)
- Proposed Methodology
- Validation
- Conclusion & Critique

Environment of UAV integrated HetNet

- Terms
  - HetNet (Heterogenous Network)
    - MBS (Macro-based Station)
    - Macro cell / Picocell / Femtocell
  - UAV (Unmanned Aerial Vehicles)
  - UE (User Equipment)
  - Intruder node



Motivation

- The collaboration of UAVs with MBS in any HetNet is desired since it can result in increasing **spectral efficiency** per unit area in dense urban scenarios or it can maximize the **coverage area** of the network.
- UAVs can be deployed for ensuring public **safety communications** keeping in view the **energy efficiency perspective**.
- The intruder/malicious nodes are able to carry out different kinds of attacks thus requiring an **optimized security technique** for the network.

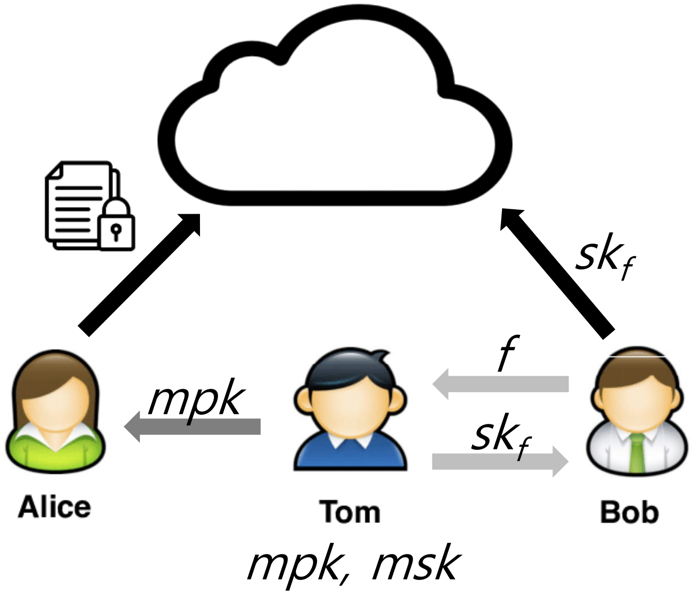
Adversary and problem statement

- Adversary: Dolev–Yao model
  - Intruder has **full control** over the network
    - overhear, intercept, and synthesize any message
  - Only limited by the constraints of the cryptographic methods used

- Problem statement
  - Ensure public safety communications on UAV integrated HetNet
  - Make the network robust against intruder attacks
  - Overcome the “all-or-nothing” disadvantage using **Functional Encryption**

Background – Functional Encryption

- Formal definition
  1) Create a master public key (*mpk*) and a master secret key (*msk*)
  2) Creates secret key (sk_f) for the function (f) using the *msk*
  3) c ← enc(*mpk*, *x*)
  4) y = $f(x)$ ← dec(sk_f, *c*)



Background – Functional Encryption

❖ **Example) Function : Inner Product**

- **Key generator (Tom)**
  1. Generate Master Keys: $\{\, Master\ Secret\ Key, Master\ Public\ Key \,\}$
  2. Already-known function: $f(\,) = [10x_1 + 8x_2 + 20x_3 + 5x_4 + \cdots]$
     - $vector\ Y = \{\, 10,\ 8,\ 20,\ 5,\ \ldots\}$
  3. Derive function key: $function\ key\ sk_f = deriveKey(Master\ Secret\ Key,\ Y)$

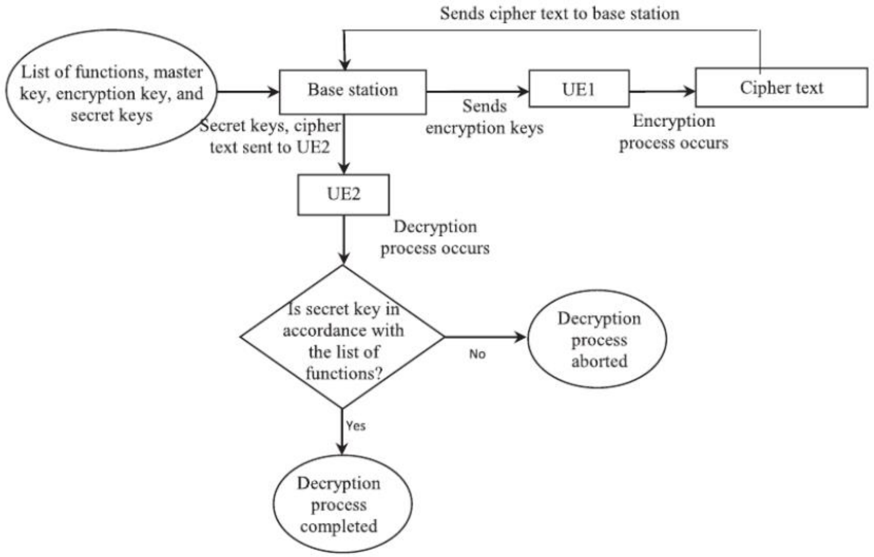
- **Data owner (Alice)**
  1. Plain data : $X = \{5,\ 2,\ 10,\ 6,\ 0,\ \ldots\}$
  2. Encrypted data: $C_X = encrypt(Master\ Public\ Key,\ X)$

- **Data user (Bob)**
  1. Receive encrypted data C_X and function key $function\ key\ sk_f$
  2. Decrypt result of $< \mathrm{X},\ Y > = decrypt(C_X,\ function\ key\ sk_f)$

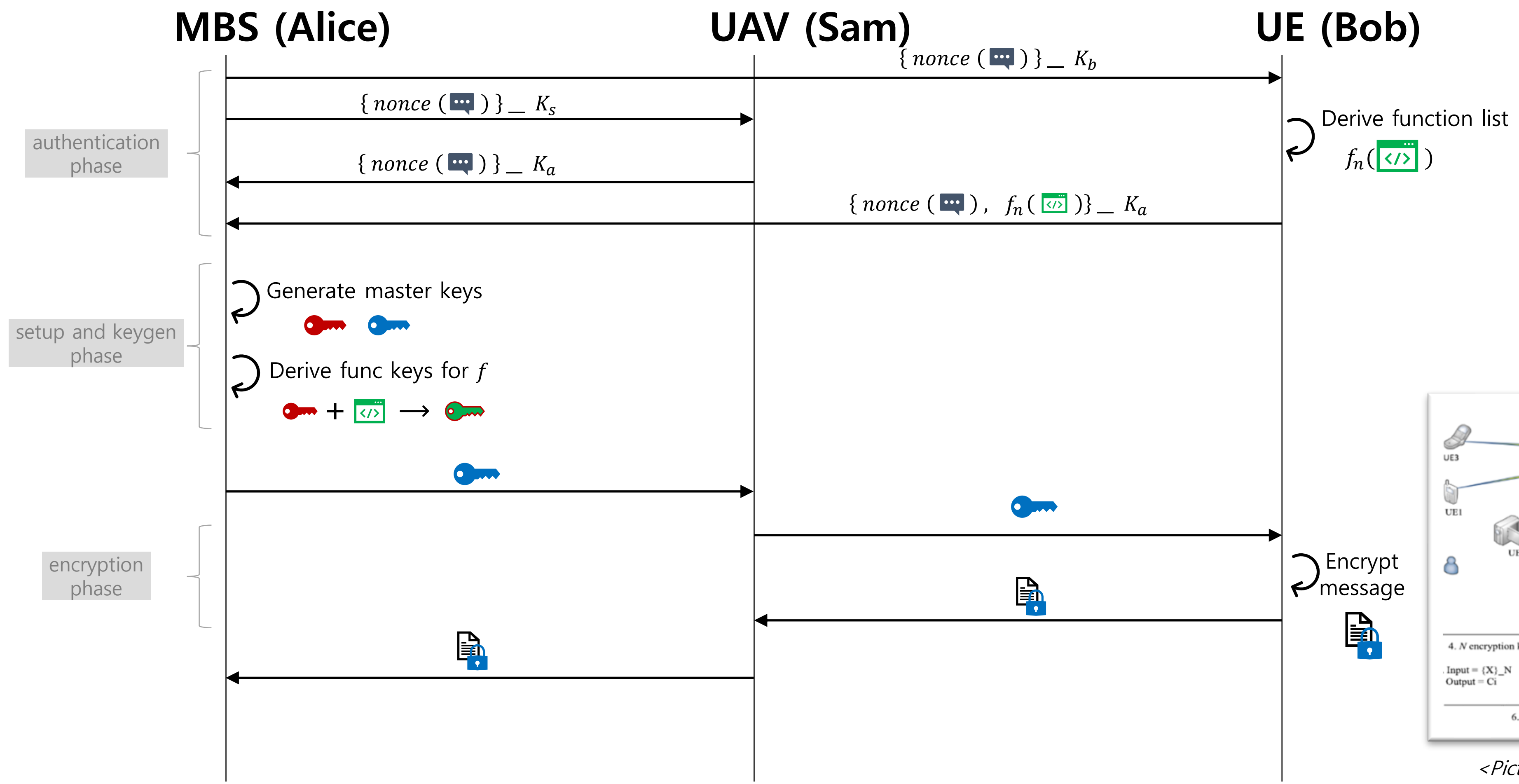
Proposed methodology

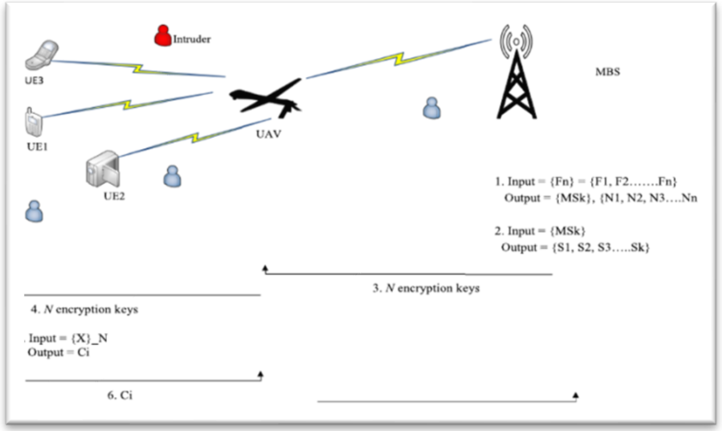
- UAV acts as a relay node for UEs which are in nonline-of-sight communication with MBS

- FE technique for securing the data transmission among UAV, UE, and MBS
  - If the intruder, somehow, is able to intercept the encrypted message, he will not be able to decrypt the whole information
  - 2 phases
    1) Between MBS and UE
    2) **Between MBS and UE through UAV**



FE between MBS and UE through UAV





<Pictorial representation of FE between MBS and UE>

Validation

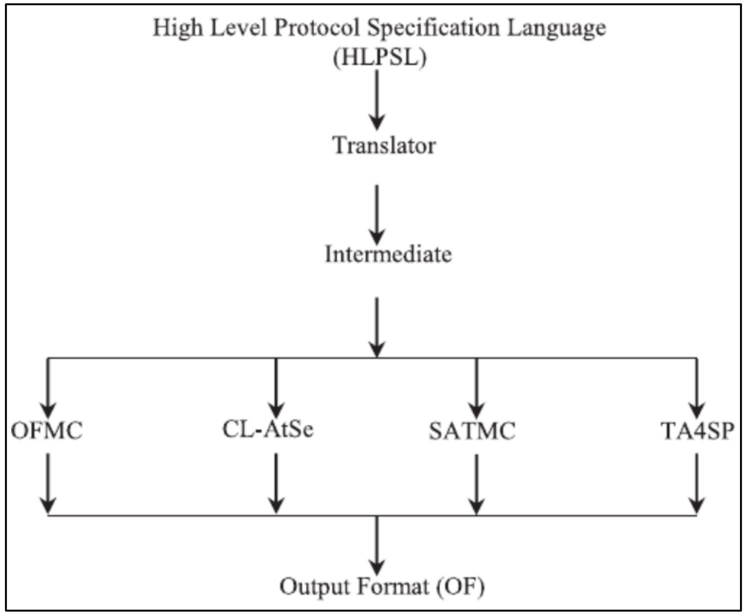
- Use the AVISPA tool which is widely used for Internet protocol validation in the early stage of development
- Run tool with protocol specification written via its language
- Compare the results of the protocol with FE and the protocol w/o FE

AVISPA tool

- **A**utomated **V**alidation of **I**nternet **S**ecurity **P**rotocol and **A**pplication
  - Widely used for detecting the vulnerabilities
  - Validate the Internet security-sensitive protocols automatically
  - Provide role-oriented, expressive, and formal language

- Modularity (4 checker modules)
  - *OFMC / CL-AtSe / SATMC / TA4SP*

- Input and output
  - In: Role-based specification (**.hlpsl file*)
  - Out : *'SAFE' / 'UNSAFE' / 'INCONCLUSIVE'*



<AVISPA tool's architecture>

How to validate (1/3)

- Example) Between MBS and UE through UAV using FE

MBS

```
role alice(MBS,UE,UAV:agent,
           Ka,Kb,Ks:public_key,
           H:hash_func,
           SND,RCV:channel(dy))
played_by MBS
def=
        local State:nat,
        Na:text,
        F,Fn,N1,Msk,N,Skf,X,Ci:message,
        Inc:hash_func
        const alice_bob_na,sec1:protocol_id
        init State := 0
        transition
            1. State=0 /\ RCV(start) =|>
               State':=1 /\ Na':=new()
                         /\ SND({MBS.UE.Na'}_Kb)
                         /\ SND({MBS.UAV.Na'}_Ks)

            2. State=1 /\ RCV({UE.MBS.Na.Fn}_Ka)
                       /\ RCV({UAV.MBS.Na}_Ka) =|>
               State':=2 /\ Msk':=new()
                         /\ N':=new()
                         /\ Skf':=new()
                         /\ N1':=H(N)
                         /\ SND({MBS.UAV.N1'}_Ks)
                         /\ request(MBS,UE,alice_bob_na,Na)
                         /\ secret({Msk,Skf},sec1,MBS)

            3. State=2 /\ RCV(UAV.MBS.Ci) =|>
               State':=3
end role
```

authentication phase

setup and keygen phase

encryption phase

UE

```
role bob(UE,MBS,UAV:agent,
         Ka,Kb,Ks:public_key,
         H:hash_func,
         SND,RCV:channel(dy))
played_by UE
def=
         local State:nat,
         Na:text,
         F,Fn,N1,Msk,N,Skf,X,Ci:message,
         Inc:hash_func
         const alice_bob_na,sec1:protocol_id
         init State := 0
         transition

            1. State=0 /\ RCV({MBS.UE.Na'}_Kb) =|>
               State':=1 /\ F':=new()
                         /\ Fn':=H(F)
                         /\ SND({UE.MBS.Na'.Fn'}_Ka)

            2. State=1 /\ RCV({UAV.UE.N1'}_Kb) =|>
               State':=2 /\ X':=new()
                         /\ Ci':={X'}_N1'
                         /\ SND(UE.UAV.Ci')

end role
```

How to validate (2/3)

- Example) Between MBS and UE through UAV using FE

UAV

```
role sam(UAV,MBS,UE:agent,
        Ka,Kb,Ks:public_key,
        H:hash_func,
        SND,RCV:channel(dy))
played_by UAV
def=
        local State:nat,
        Na:text,
        F,Fn,N1,Msk,N,Skf,X,Ci:message,
        Inc:hash_func
        const alice_bob_na,sec1:protocol_id
        init State := 0
        transition
```

authentication phase

```
            1. State=0 /\ RCV({MBS.UAV.Na'}_Ks) =|>
            State':= 1 /\ SND ({UAV.MBS.Na}_Ka)

            2. State=1 /\ RCV({MBS.UAV.N1'}_Ks) =|>
             State':=2 /\ SND({UAV.UE.N1}_Kb)

            3. State=2 /\ RCV(UE.UAV.Ci') =|>
             State':=3 /\ SND(UAV.MBS.Ci)
end role
```

relay phase

Envs.

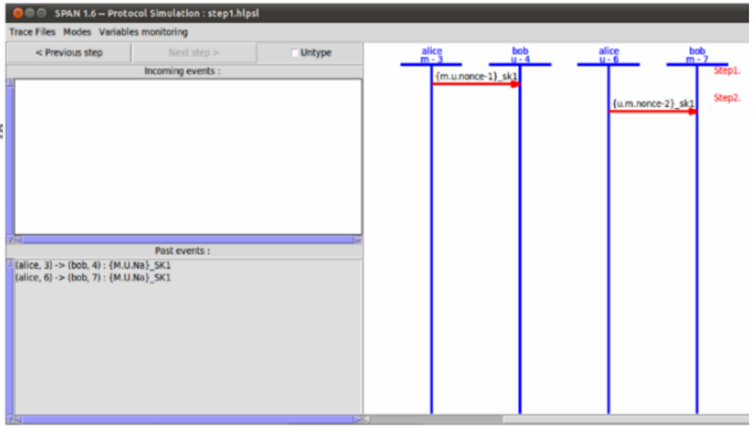
```
role session(MBS,UE,UAV:agent,
             Ka,Kb,Ks:public_key,
             H:hash_func)
def=
        local
                SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy)
        composition
                alice(MBS,UE,UAV,Ka,Kb,Ks,H,SND1,RCV1)
                /\ bob(UE,MBS,UAV,Ka,Kb,Ks,H,SND2,RCV2)
                /\ sam(UAV,MBS,UE,Ka,Kb,Ks,H,SND3,RCV3)
end role

role environment()
def=
        const
                mbs,ue,uav:agent,
                ka,kb,ks:public_key,
                h:hash_func,
                na,f,fn,n1,msk,n,skf,x,ci:message,
                alice_bob_na,sec1:protocol_id
        intruder_knowledge = {mbs,ue,uav,ka,kb,ks,na,f,fn,n1,msk,n,skf,x,ci}
        composition
                session(mbs,ue,uav,ka,kb,ks,h)
                /\ session(ue,mbs,uav,ka,kb,ks,h)
                /\ session(uav,mbs,ue,ka,kb,ks,h)
end role
goal
        secrecy_of sec1
      authentication_on alice_bob_na
end goal
environment()
```

How to validate (3/3)

- Output of AVISPA tool

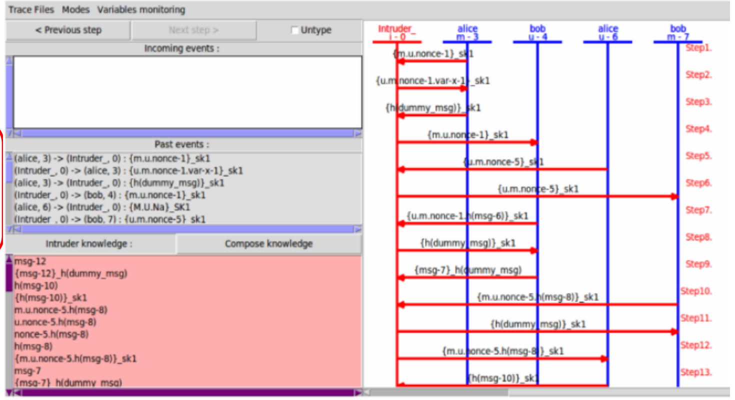
SAFE

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/step1.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.02s
  visitedNodes: 25
  depth: 4 plies
```



UNSAFE

```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/span/span/testsuite/results/step1unsafe.if
GOAL
  authentication_on_alice_bob_na
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.05s
  visitedNodes: 1 nodes
  depth: 1 plies
ATTACK TRACE
i -> (m,3): start
(m,3) -> i: {m.u.Na(1)}_sk1
i -> (m,3): {u.m.Na(1).x254}_sk1
(m,3) -> i: {h(dummy_msg)}_sk1
```



Validation result

- *It sounds obvious, but AVISPA tool says using FE technique is safe..*

Comparison	Protocol	Result
Without using FE	Between MBS and UE	UNSAFE
	Between MBS and UE through UAV	UNSAFE
With using FE	Between MBS and UE	SAFE
	Between MBS and UE through UAV	SAFE

Conclusion (wrap-up)

- UAV integrated HetNet is vulnerable so need to provide security to the entire network and the data

- This article has proposed **FE technique to the UAV integrated HetNet** in two phases
  - FE between UE and MBS
  - FE between UE and MBS through UAV

- This approach has been validated by AVISPA tool and provides the desired security to the UE and its data

Critiques

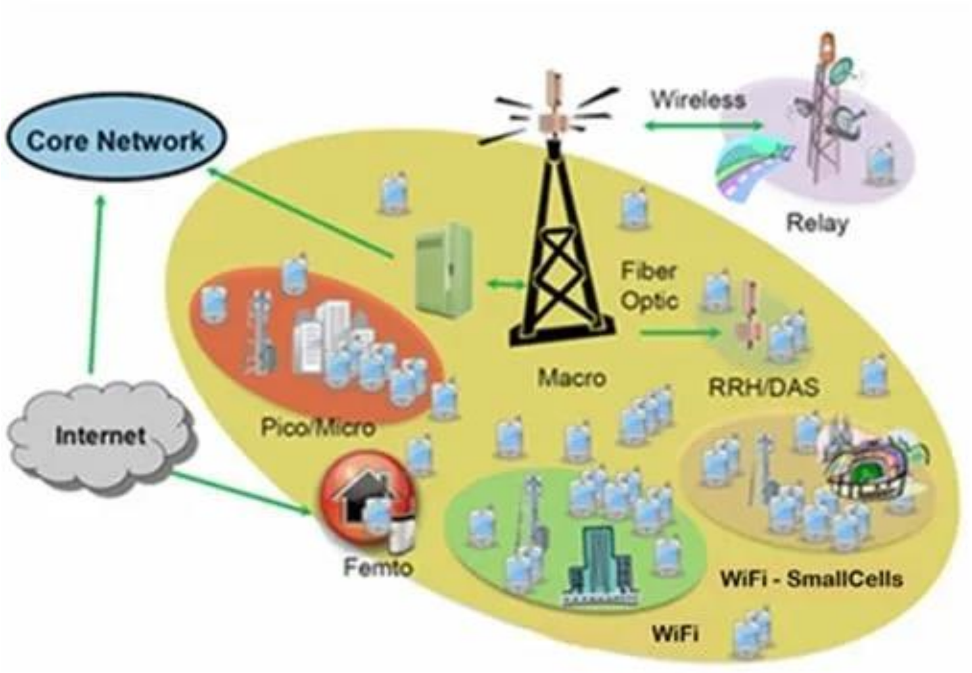
- Good things
  - Considering UAVs as relay nodes *(not initiated from this paper though)*
  - New utilization of FE technique *(and AVISPA)*
- Bad things
  - Lack of real-world use scenario
    - Unclear function or data format
  - Too simple comparison and experiment
    - Only FE vs non-FE
  - Limited scope
    - e.g. routing, performance, or energy(efficiency) issues

Thank you

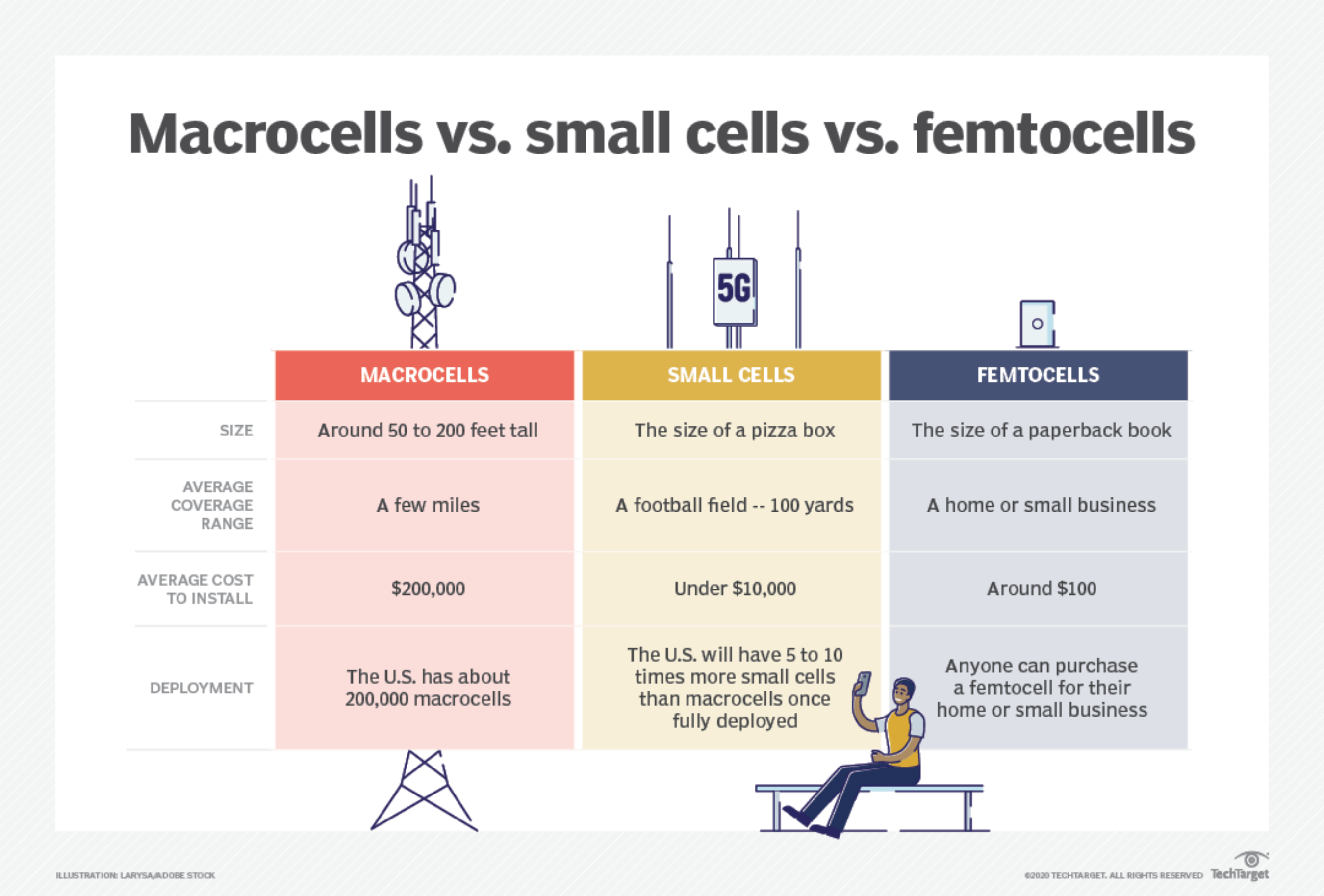
Appendix

Appendix #1 HetNet

- HetNet describes wireless networks using different access technologies
  - For example, a wireless network that provides a service through a wireless LAN and is able to maintain the service when switching to a cellular network is called a wireless heterogeneous network.
  - A Wide Area Network can use some combination of macrocells, picocells, and femtocells in order to offer wireless coverage in an environment with a wide variety of wireless coverage zones.



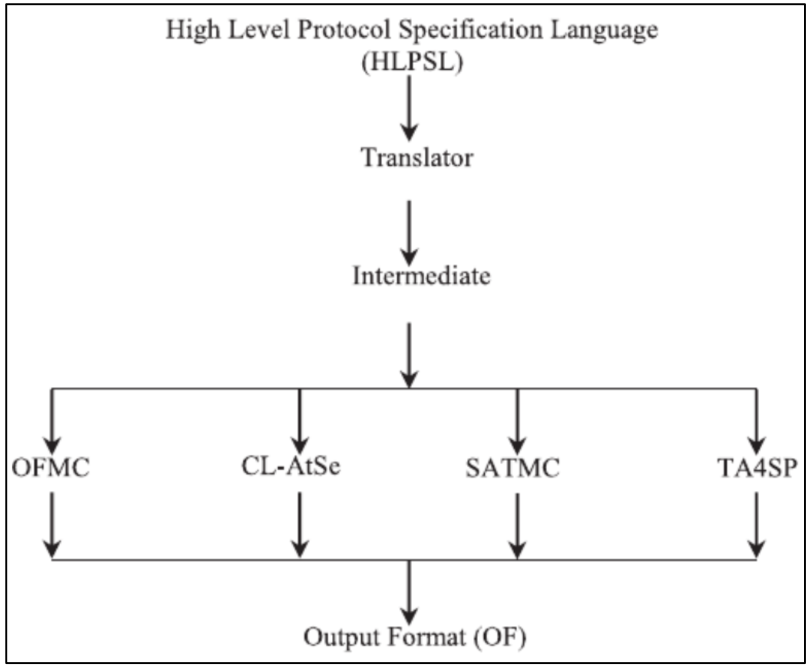
Appendix #2 macro cell



Macro base stations are **fundamental elements in any heterogeneous network** (HetNet) wireless infrastructure to provide coverage and support capacity.

Appendix #3 AVISPA

- **A**utomated **V**alidation of **I**nternet **S**ecurity **P**rotocols and **A**pplications.
  - http://www.avispa-project.org



- Modules
  - OFMC
    - On-the-fly model checker
  - CL-AtSe
    - Constraint logic-based attack searcher
  - SATMC
    - SAT-based model checker
  - TA4SP
    - Tree automata based on automatic approximations for the analysis of security protocols

Appendix #4 vs. other techniques

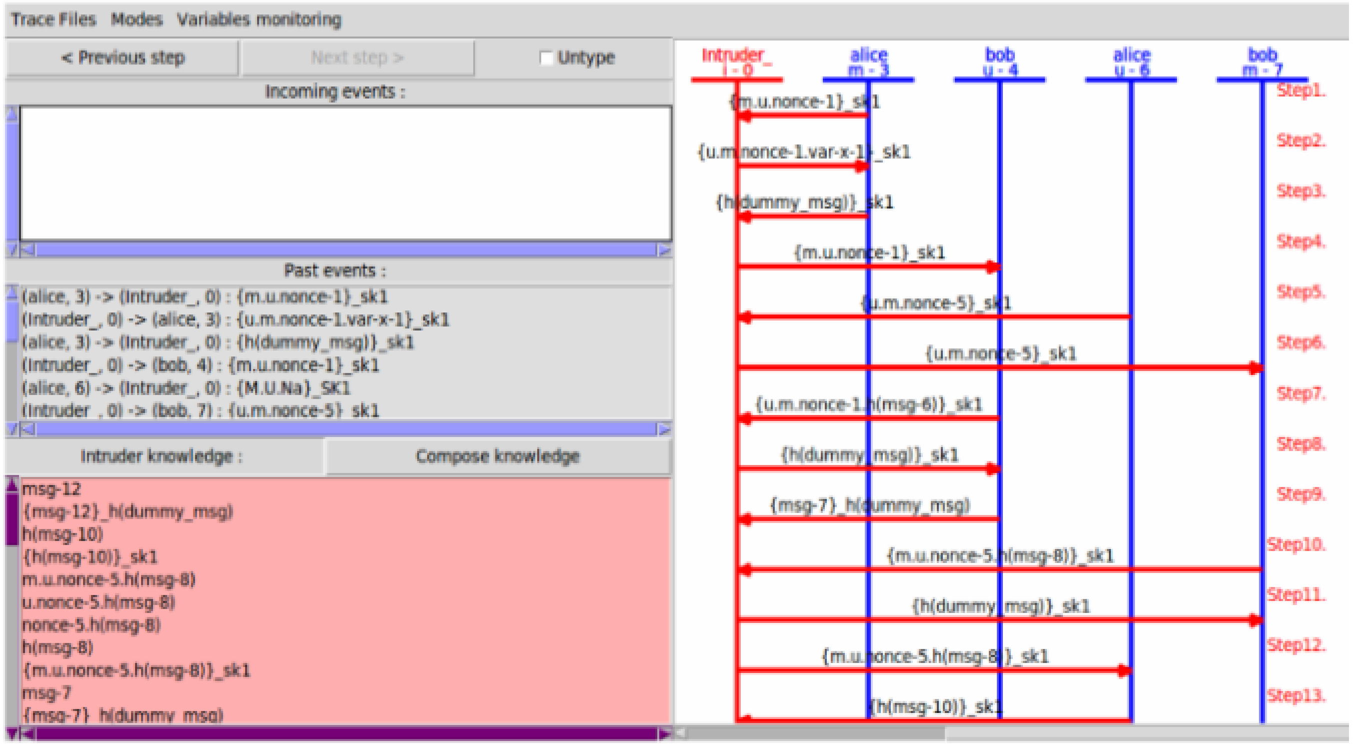
- ABE / IBE
- DH key exchange
- Homomorphic Encryption

TABLE 4 Comparative study between the conventional approaches and the proposed scheme

Conventional approaches			Proposed scheme: Functional encryption
Reference	**Techniques**	**Demerits**	**Merits**
18	Attribute-based cryptography technique	During decryption process, the ciphertext is decrypted wholly which reveals the entire message.	During decryption process, the ciphertext is decrypted in particular portions only, for which secret keys are present.
19	Identity-based encryption	If public key generation center is compromised, then the data is at a greater risk of disclosure.	In MBS, entire data has been stored in accordance with list of functions and the secret keys are generated, respectively, for each function of plaintext. So, if MBS is compromised, even then, entire data is not at a risk of disclosure.
21	Diffie-Hellman key exchange	Authentication process is not done.	Authentication process is done among all entities by using nonce signal.
23	Homomorphic encryption	The decryption of ciphertext will either reveal the entire message or will not be decrypted at all.	The decryption of ciphertext will generate only that portion of plaintext which the user has demanded.
44	Challenge-based trust mechanism	Strategy can be failed, if the malicious nodes have some information about nearby traffic.	Even if the malicious nodes have some information regarding ongoing traffic, they cannot reveal the entire plaintext message as it is encrypted according to different functions.

Result #1

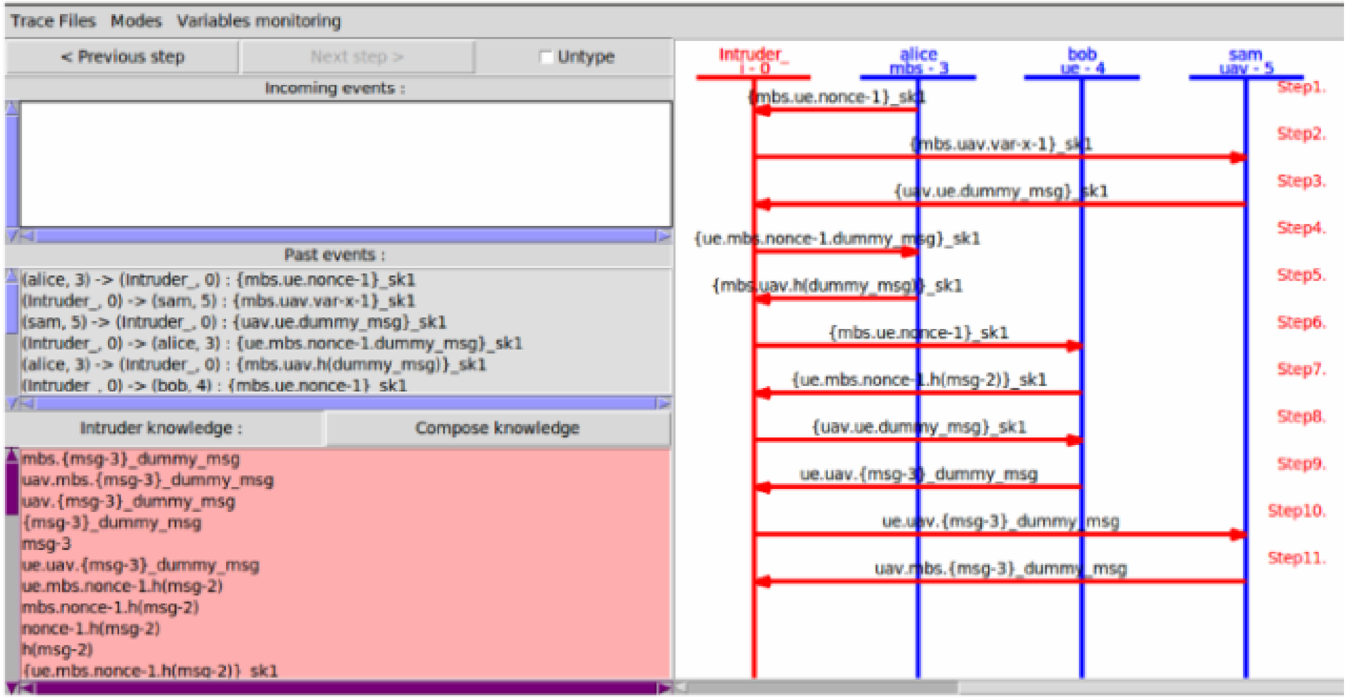
- Between MBS and UE w/o FE




```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/span/span/testsuite/results/step1unsafe.if
GOAL
  authentication_on_alice_bob_na
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.05s
  visitedNodes: 1 nodes
  depth: 1 plies
ATTACK TRACE
i -> (m,3): start
(m,3) -> i: {m.u.Na(1)}_sk1
i -> (m,3): {u.m.Na(1).x254}_sk1
(m,3) -> i: {h(dummy_msg)}_sk1
```

Result #2

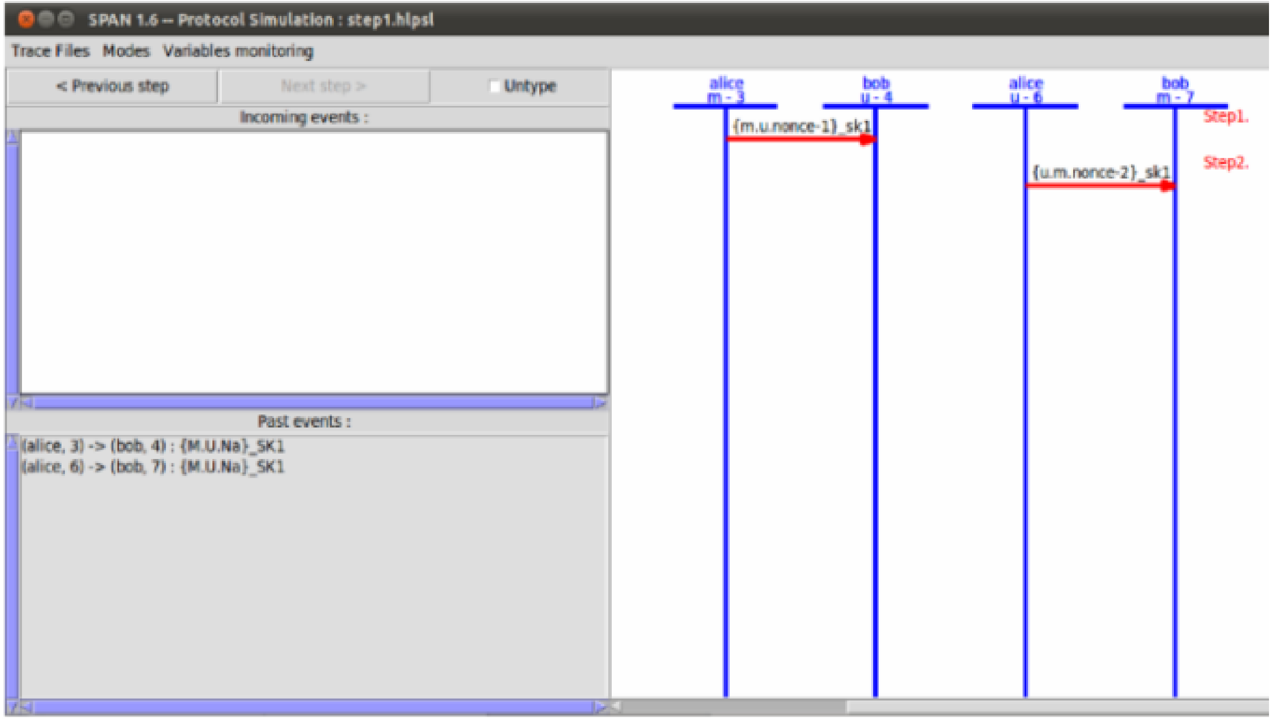
- Between MBS and UE through UAV w/o FE




```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/span/span/testsuite/results/step2unsafe.if
GOAL
  authentication_on_alice_bob_na
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.08s
  visitedNodes: 6 nodes
  depth: 2 plies
ATTACK TRACE
i -> (mbs,3): start
(mbs,3) -> i: {mbs.ue.Na(1)}_sk1
i -> (uav,3): {mbs.uav.x255}_sk1
(uav,3) -> i: {uav.ue.dummy_msg}_sk1
i -> (mbs,3): {ue.mbs.Na(1).dummy_msg}_sk1
(mbs,3) -> i: {mbs.uav.h(dummy_msg)}_sk1
```

Result #3

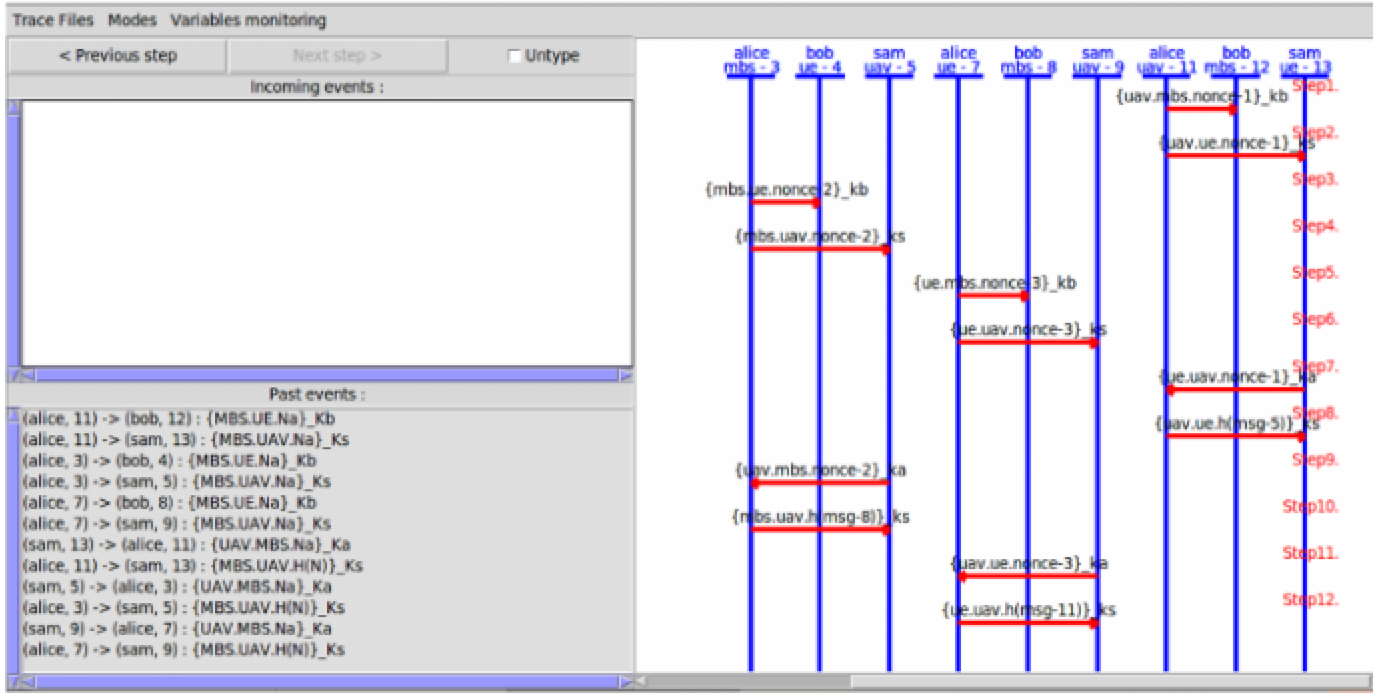
- Between MBS and UE w/ FE




```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/step1.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.02s
  visitedNodes: 25 nodes
  depth: 4 plies
```

Result #4

- Between MBS and UE through UAV w/ FE




```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/step2.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.02s
  visitedNodes: 63 nodes
  depth: 6 plies
```