An Efficient Protocol for UAS Security

Olivier Blazy[1], Pierre-François Bonnefoi[1], Emmanuel Conchon[1], Damien Sauveron[1,3]
Raja Naeem Akram[2], Konstantinos Markantonakis[2], Keith Mayes[2], **Serge Chaumette**[3]

1: XLIM (UMR CNRS 7252), MATHIS, Université de Limoges

2: Information Security Group Smart Card Centre, Royal Holloway, University of London

3: LaBRI (UMR CNRS 5800), Université de Bordeaux

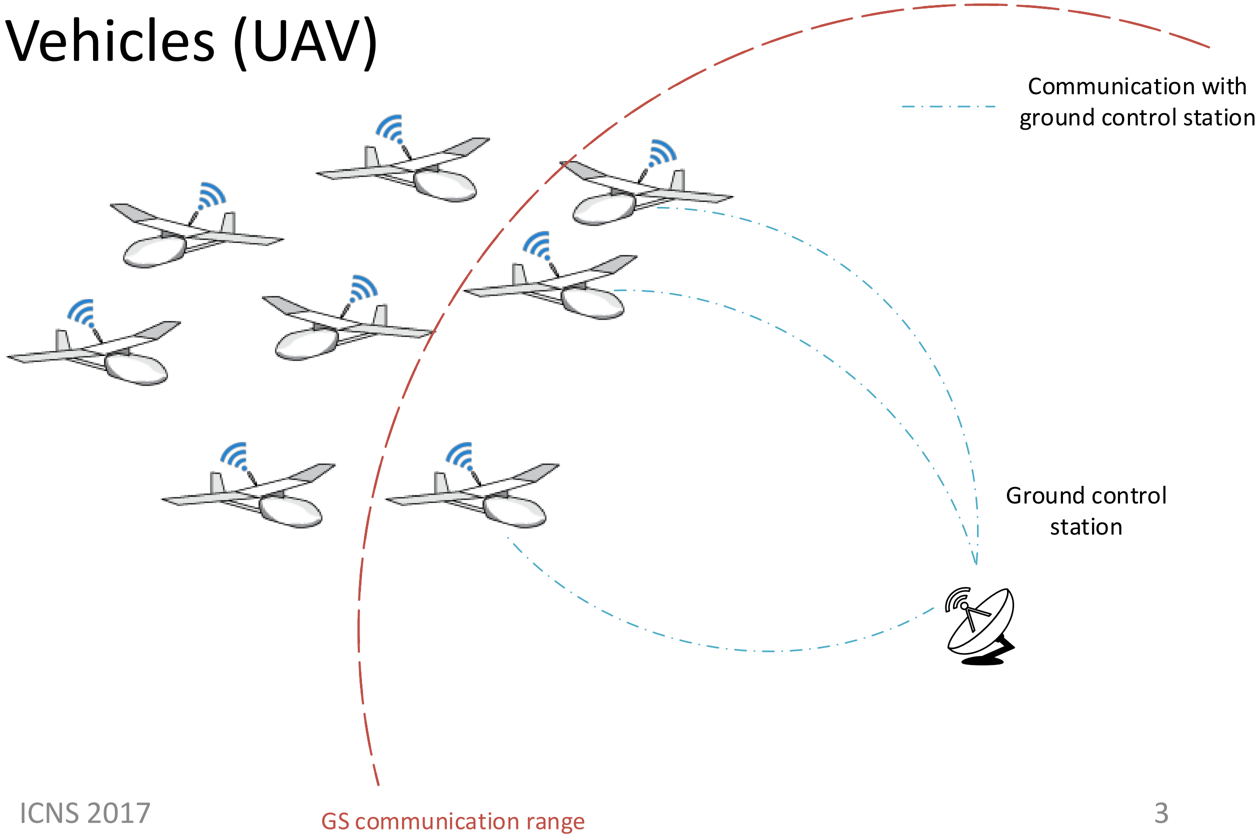
Roadmap

- Introduction
- Contributions
- Requirements
- Cryptographic techniques used
- Protocol
  - Pre-protocol Setup
  - UAV processes
  - Protocol
- Formal Proof & Analysis of Efficiency - Test-bed
- Conclusions and Future work

Introduction

- Unmanned Aerial Systems (UAS)
  - Ground Control Station (GCS or GS)
  - One or several Unmanned Aerial Vehicles (UAV)

- UAVs sense and store data
- UAVs send data to GS when communication is possible (UAVs in the range)



Introduction

- Attacker interests in UAS

BBC | Sign in | News | Sport | Weather | Shop | Earth | Travel

NEWS

Home | Video | World | UK | Business | Tech | Science | Magazine | Entertainment & Arts

World | Africa | Asia | Australia | Europe | Latin America | Middle East | US & Canada

Iran shows 'hacked US spy drone' video footage

7 February 2013 | Middle East

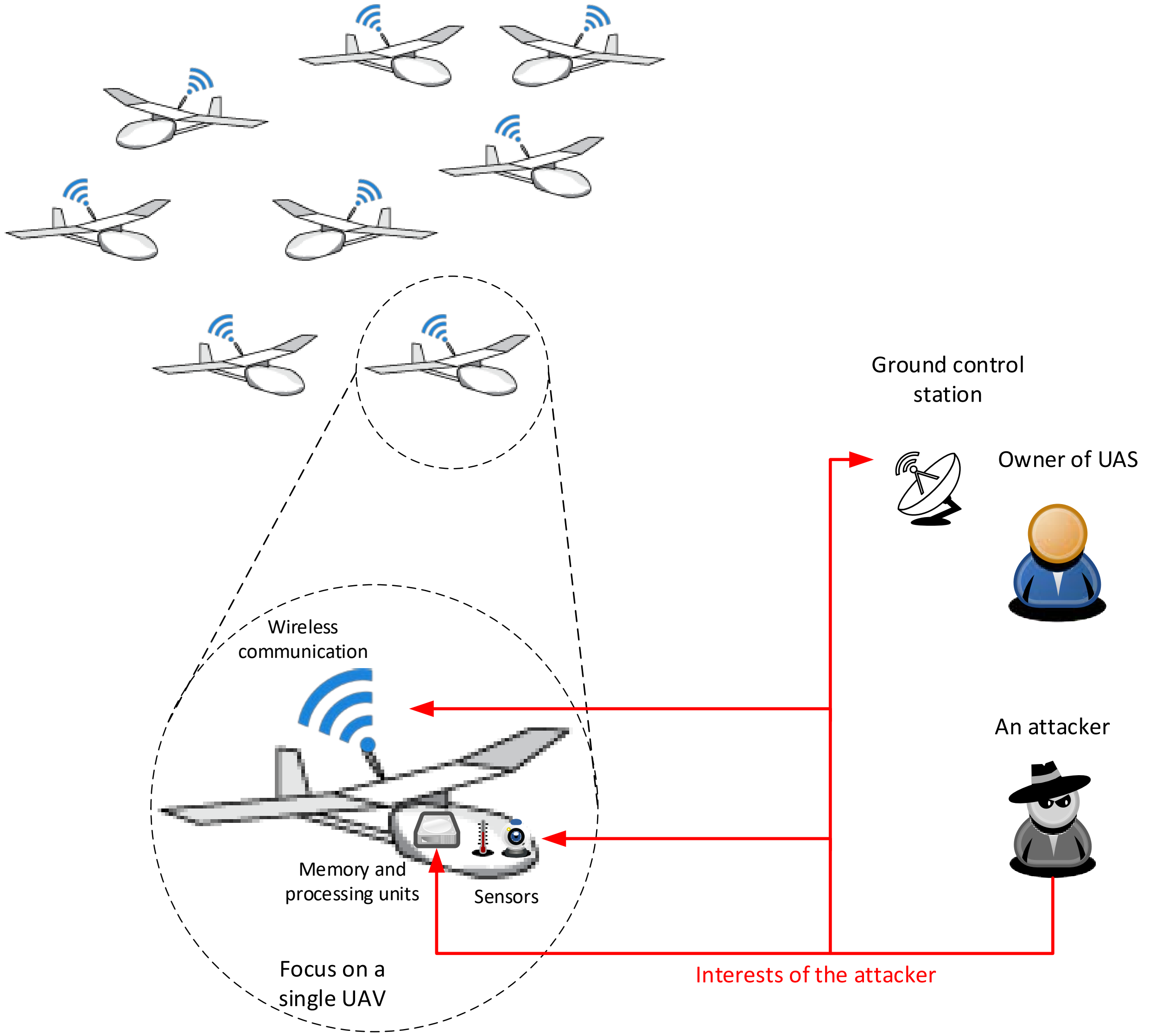
Share

Iran has released what it says is decoded video footage extracted from a US surveillance drone captured in 2011.

The material broadcast on Iranian state television purports to show a US base and the Afghan city of Kandahar. It is not clear if the footage is genuine.

The video footage has not been verified

Last year Iran said it was building a copy of the drone - an RQ-170 Sentinel - after breaking its encryption codes.

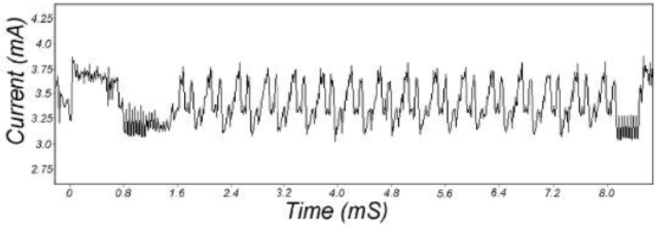


Introduction

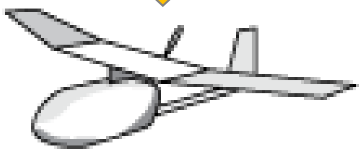
- We consider a strong adversary model with a high attack potential.
  - the adversary has capabilities and knowledge to capture a UAV in a functional state

Then, he can perform advanced attacks

SPA on DES ciphering

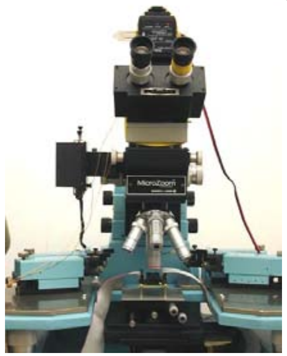


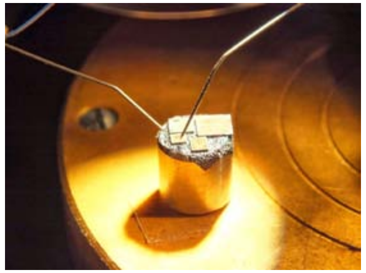
Side-channel attacks



Fault injection attacks

Physical attacks





Contributions

- An Efficient Protocol for UAS Security
  - To ensure confidentiality of sensed data
    - using efficient cryptographic techniques (encryption scheme is left to implementer choice)
    - withstanding an adversary with a high attack potential
  - To minimize exchanges between UAVs and GS
    - 1 round is required (except in an optional case: 1.5 rounds).
- A Formal Proof of the Proposed Protocol

Requirements

- Each UAV must have its own cryptographic means (keys)
  - In other words, capture and forensic of UAVs should not compromise the security of UAS
- Keys must evolve during the mission to ensure the Perfect Forward and Backward Secrecy properties
- Cryptographic means of UAV should be renewed/refreshed from time to time
  - The C2 links can be used to refresh them
- Collected (sensed) data must be sent to the Ground Station as soon as a connection is possible to avoid potential loss

- Assumption: The GS is secure (else the whole network would be corrupted).

Cryptographic Techniques Used

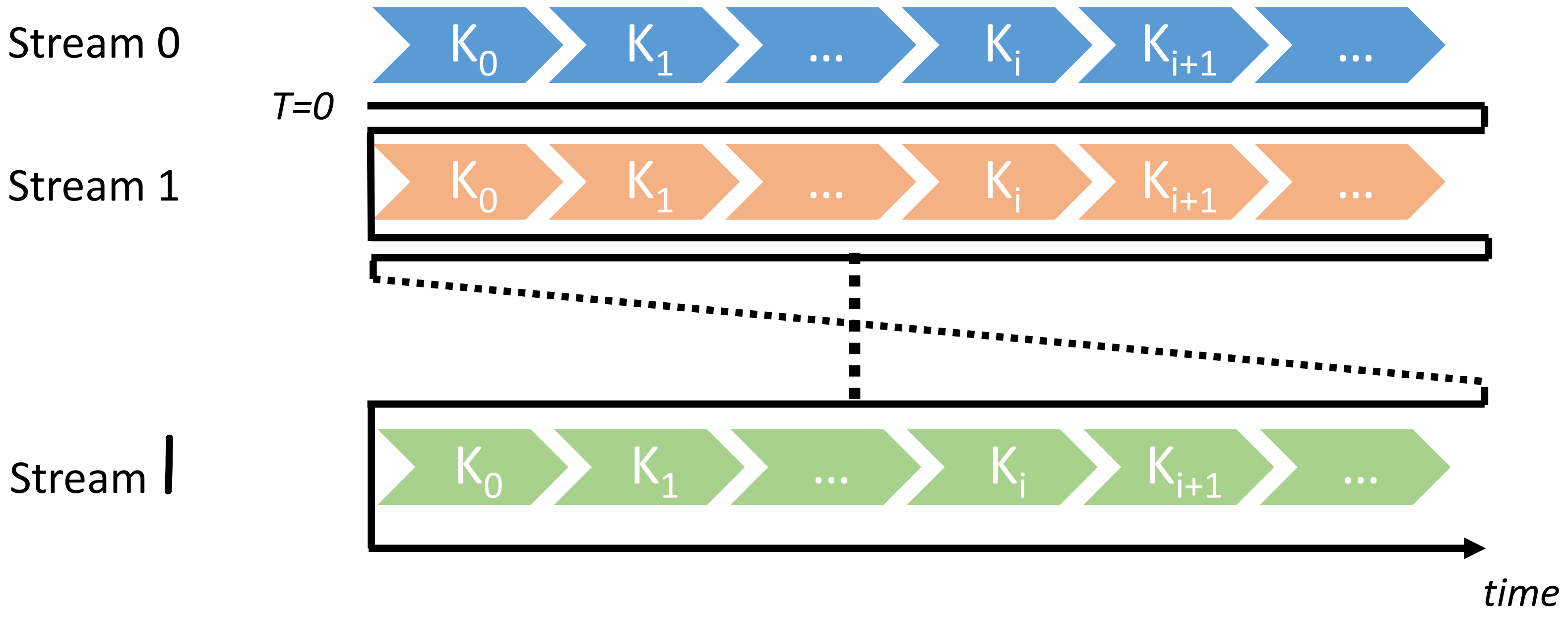
- Keys stream
  - Based on an origin (the first key)
  - Subsequent keys are generated using a function (and potential parameters to diversify the result)

  - We use a keyed hash function diversified with ID of UAV

$$K_{i+1} = H_{\mathsf{UAV}_{\mathsf{ID}}}(K_i)$$

Cryptographic Techniques Used

- Keys streams are timely updated to prevent attacks (since it is well known that an attacker can find subsequent keys in a stream if he knows only one key



Cryptographic Techniques Used

- One–time key: each key is used only once to encrypt data
  - The key is used:
    - to encrypt data
    - to compute a triplet of Authentication Tickets (used latter in the protocol for C2)

$$(H_1, H_2, H_3)$$

$$H_1 = H(K_i \| 1)$$
$$H_2 = H(K_i \| 2)$$
$$H_3 = H(K_i \| 3)$$

    - to generate the subsequent key of the stream

  - Then, the key is cleared from memory and it cannot be recovered by anyone

Protocol Notations

UAV	:	Denotes an Unmanned Aerial Vehicle.
GS	:	Denotes a Ground Station.
$A \rightarrow B$	:	Message sent by an entity A to an entity B.
X_{ID}	:	Represents the identity of an entity X.
$X \| Y$	:	Represents the concatenation of the data items X, Y in the given order.
$X \oplus Y$	:	Represents the xor operation of the data items X, Y.
$[D]^k$	:	Data D are encrypted by a one-time key k.
$H(Z)$	:	Is the result of generating a hash of data Z.
$H_k(Z)$	:	Result of generating a keyed hash of data Z using key k.
$K^{\ell}_{\mathsf{UAV_{ID}}}$	:	The ℓ^{th} keys stream origin. This key is randomly chosen to initialize the ℓ^{th} stream of keys used to encrypt the sensed data. It is generated by the GS and sent to UAV $\mathsf{UAV_{ID}}$. In the pre-protocol setup, $K^{0}_{\mathsf{UAV_{ID}}}$ is set by the GS in UAV $\mathsf{UAV_{ID}}$.
K_i	:	A one-time key which evolves at each encryption of sensed data. The first key, K_0 is initialized using the value of the current keys stream origin $K^{\ell}_{\mathsf{UAV_{ID}}}$. Subsequent keys are computed with $K_{i+1} = H_{\mathsf{UAV_{ID}}}(K_i)$
SD_j	:	Denotes the j^{th} block of sensed data.
H_1	:	Denotes the following computation $H(K_i \| 1)$.
H_2	:	Denotes the following computation $H(K_i \| 2)$.
H_3	:	Denotes the following computation $H(K_i \| 3)$.
i_{lastKS}	:	Denotes the rank of the last key used in the previous keys stream.
Command	:	Denotes any command from the GS to UAV. Two examples of command are: 1) ACK to inform UAV that data have been received by GS and then can be deleted from its internal non-volatile memory. 2) NKS to inform the UAV to change the keys stream origin to $K^{\ell+1}$.
$\mathsf{Command}_{ack}$	:	Denotes an Acknowledgment to some commands by UAV. An example of such acknowledgment is for the NKS command for which the UAV informs the GS of the last K_i of the current keys stream used to encrypt the sensed data.

Pre-Protocol Setup

- Each UAV is preconfigured with origin of its first keys stream

$$K_0 = K^0_{\mathsf{UAV}_{\mathsf{ID}}}$$

- The GS is pre-configured with the first keys stream for each UAV of the UAS

UAV in Mission – Sensing & encryption Process

- Each sensed data block SD_j is immediately encrypted and then stored in non-volatile memory of the UAV using the current key, K_i
  - SD_j is encrypted with any efficient symmetric algorithm using K_i and the result $[SD_j \,||\, UAV_{ID}]^{K_i}$ is stored in NVM
    - UAV_{ID} is added to encrypted data to allow the GS to verify the result has meaning when coming from the UAV
- For each above encryption, UAV must also compute and store the triplet of Authentication ticket (H_1, H_2, H_3)
  - These tickets will be used later to decrypt commands on C2 link.

$$H_1 = H(K_i \| 1)$$
$$H_2 = H(K_i \| 2)$$
$$H_3 = H(K_i \| 3)$$

- The subsequent key K_{i+1} is computed and the current one, K_i, is deleted from memory

$$K_{i+1} = H_{\mathsf{UAV_{ID}}}(K_i)$$

UAV in Mission – Communication Process

- When UAV is in communication range of GS, it sends available encrypted data: $[SD_j \,||\, UAV_{ID}]^{K_i}$, …, $[SD_{j+n} \,||\, UAV_{ID}]^{K_{i+n}}$ and keeps them until it receives an authenticated command from GS
  - One authenticated command is required by encrypted SD. If UAV does not received the related authenticated command, it will send these encrypted data again and again until it receives it.

- When UAV receives commands from GS, it authenticates them with the computed Authentication ticket (H_1, H_2, H_3): it can then delete from its memory the encrypted data acknowledged along with the triplet related to the ticket used to authenticate the command.
  - There are 3 types of commands:
    - The ACK command is only used by GS to acknowledge receipt of data
    - The NKS command is to change the key stream to a new one. The new origin is provided along with the command.
      - Note to avoid some desynchronization attacks, for this specific command the UAV has to acknowledge it has change of keys stream
    - Other commands can be normal C2 commands.

UAV to GS Secure Communication Protocol

1.	UAV $\rightarrow$ GS	:	$\mathsf{UAV}_{\mathsf{ID}} \Vert [\mathsf{SD}_j \Vert \mathsf{UAV}_{\mathsf{ID}}]^{K_i}$
2.	GS $\rightarrow$ UAV	:	$\mathsf{UAV}_{\mathsf{ID}} \Vert \mathsf{Command}$
		:	with $\mathsf{Command} = H_1 \oplus \mathsf{ACK}$ for ACK
		:	with $\mathsf{Command} = H_2 \oplus (\mathsf{NKS} \Vert K^{\ell+1}_{\mathsf{UAV}_{\mathsf{ID}}})$ for New Keys Stream
		:	with $\mathsf{Command} = H_3 \oplus (< \mathit{any\ command} >)$ for any other command
3.	UAV $\rightarrow$ GS	:	$\mathsf{UAV}_{\mathsf{ID}} \Vert \mathsf{Command}_{ack}$
	(optional step)	:	with $\mathsf{Command}_{ack} = [\mathsf{ACK}_{\mathsf{NKS}} \Vert i_{lastKS}]^{K_0}$ with $K_0 = K^{\ell+1}_{\mathsf{UAV}_{\mathsf{ID}}}$

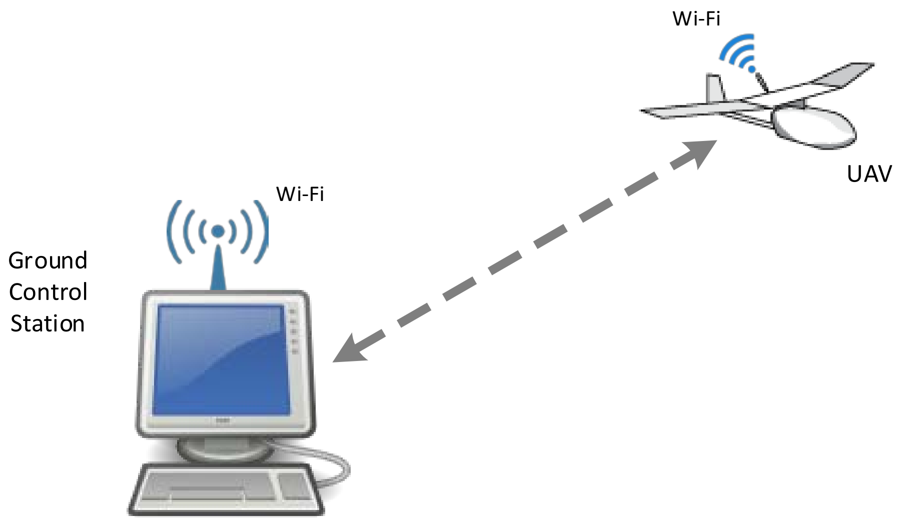
Formal Proof & Analysis of Efficiency

- Using security experiments, in the random oracle model, we have proven that the proposed protocol is secure under the security of the chosen encryption scheme.

- Most operations used in the protocol are lightweight: xor, hash function, keyed hash function
- The only not lightweight operation is the chosen encryption scheme, denoted by [ ], whose choice is left free to implementer.

Test-bed for UAS

- The UAV is a Parrot AR.Drone2 running Linux
  - Encryption scheme chosen is AES
  - Hash and keyed-hash functions are based on SHA-256
- The Ground Control Station is a desktop computer with a Wi-Fi card.



Conclusions and Future work

- Our protocol for UAS is efficient and secure against an attacker with a high attack potential.
- In addition, it is flexible: implementer can choice the encryption scheme

- We plan to extend it to hierarchical UAS
  - Several GSs
  - Network with big UAV acting as cluster head

Acknowledgements to

- the **SFD** (**S**ecurity of **F**leets of **D**rones) project
  - funded by Region Limousin;
- the **TRUSTED** (**TRUS**ted **TE**stbed for **D**rones) project
  - funded by the CNRS INS2I institute through the call 2016 PEPS (*"Projet Exploratoire Premier Soutien"*) SISC (*"Securité Informatique et des Systèmes Cyber-physiques"*);
- the **SUITED** (**S**uited sec**U**r**I**ty **TE**stbed for **D**rones), **SUITED2** and **UNITED** (**U**nited **N**etwork**I**ng **TE**stbed for **D**rones), **UNITED2**
  - projects funded by the MIRES (Mathematiques et leurs Interactions, Images et information numérique, Réseaux et Sécurité) CNRS research federation;
- the **SUITED-BX** and **UNITED-BX** projects
  - funded by LaBRI and its MUSe team.

Thank You!
Any Questions or Suggestions

Backup slide for Security Experiment

$\mathsf{Exp}_{\mathcal{E},\mathcal{A}}^{\mathsf{drone}-b}(1^n)$

1. $(K^0_{\mathsf{UAV}_{\mathsf{ID}}})_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(1^n)$
2. $(M_0, M_1, \mathsf{ID}^*, k^*, \ell^*) \leftarrow \mathcal{A}(\texttt{FIND} : \mathsf{OCorrupt}(\cdot,\cdot,\cdot))$
3. $C^* \leftarrow \mathsf{Encrypt}(K^{k^*,\ell^*}_{\mathsf{UAV}_{\mathsf{ID}^*}}, M_b)$
4. $b' \leftarrow \mathcal{A}(\texttt{GUESS} : C^*, \mathsf{OCorrupt}(\cdot,\cdot,\cdot))$
5. $\texttt{IF}\ (\mathsf{ID}^*, k^*, \ell^*) \tilde{\in} \mathsf{CS}\ \texttt{RETURN}\ \bot$
6. $\texttt{ELSE}\ \texttt{RETURN}\ b'$