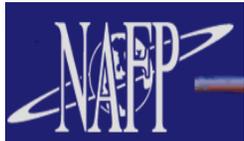


An Elliptic Curve Based Authentication Protocol For Controller-Pilot Data Link Communications

Dawit Getachew, PhD
Chicago State University

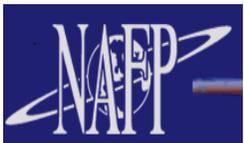
&

James H. Griner Jr.
NASA Glenn Research Center



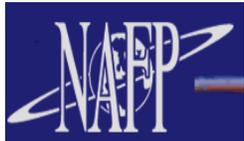
OBJECTIVE

- To perform a simulation study to look into processing delay and overhead due to implementation of an authentication protocol to CPDLC messages between an aircraft and ground system.
- To carry out the objective we broke the problem into two stages:
 - Pre-software stage  Model specification
 - Software stage  Utilization of simulation package

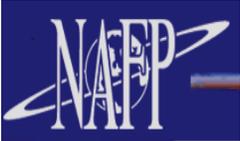
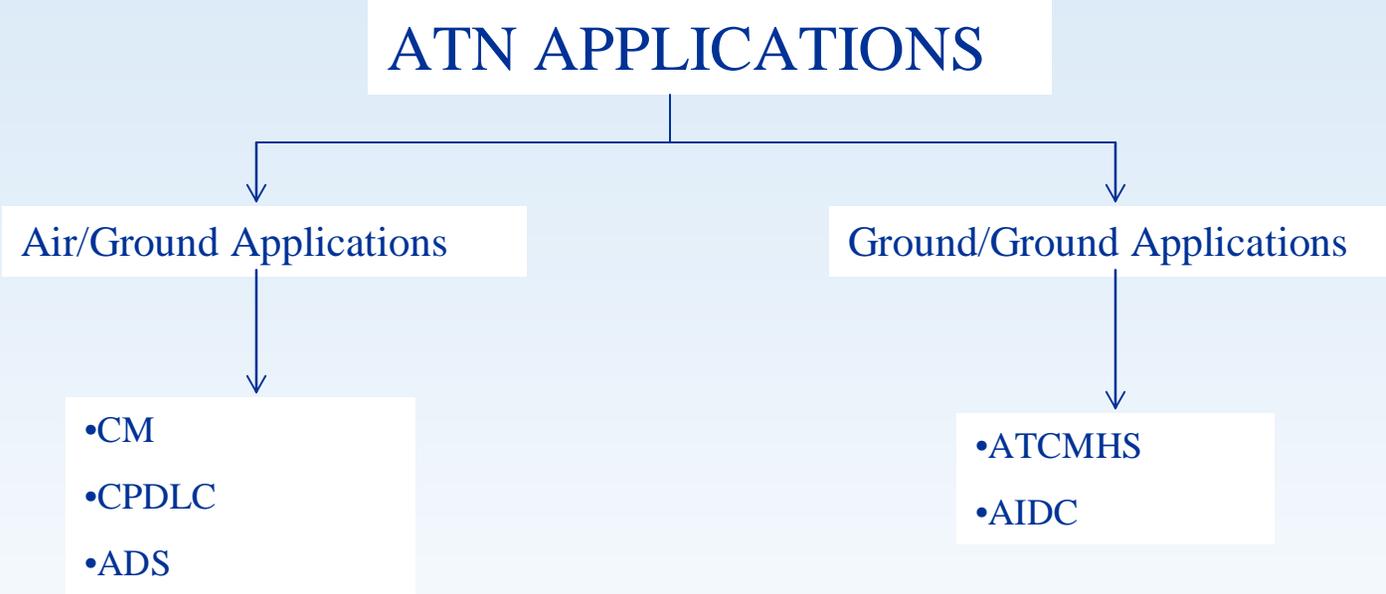


Presentation Topics

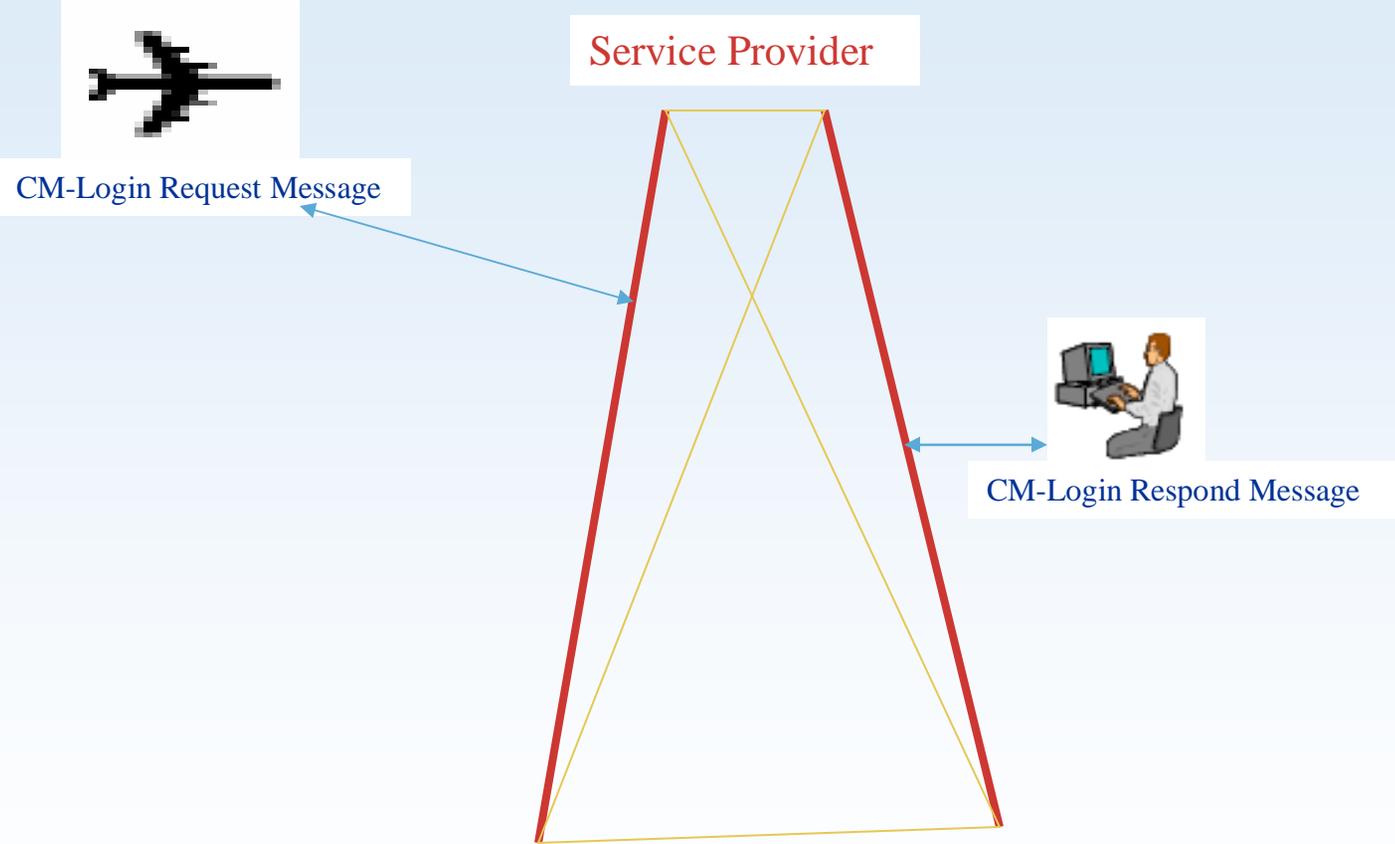
- ATN Application Architecture
- ATN Security Requirement
- Elliptic Curve Cryptography
 - Primitives
 - Schemes
- Proposed Authentication Protocol
 - Shared Public Value
 - Authentication Protocol
- Conclusion
- Question



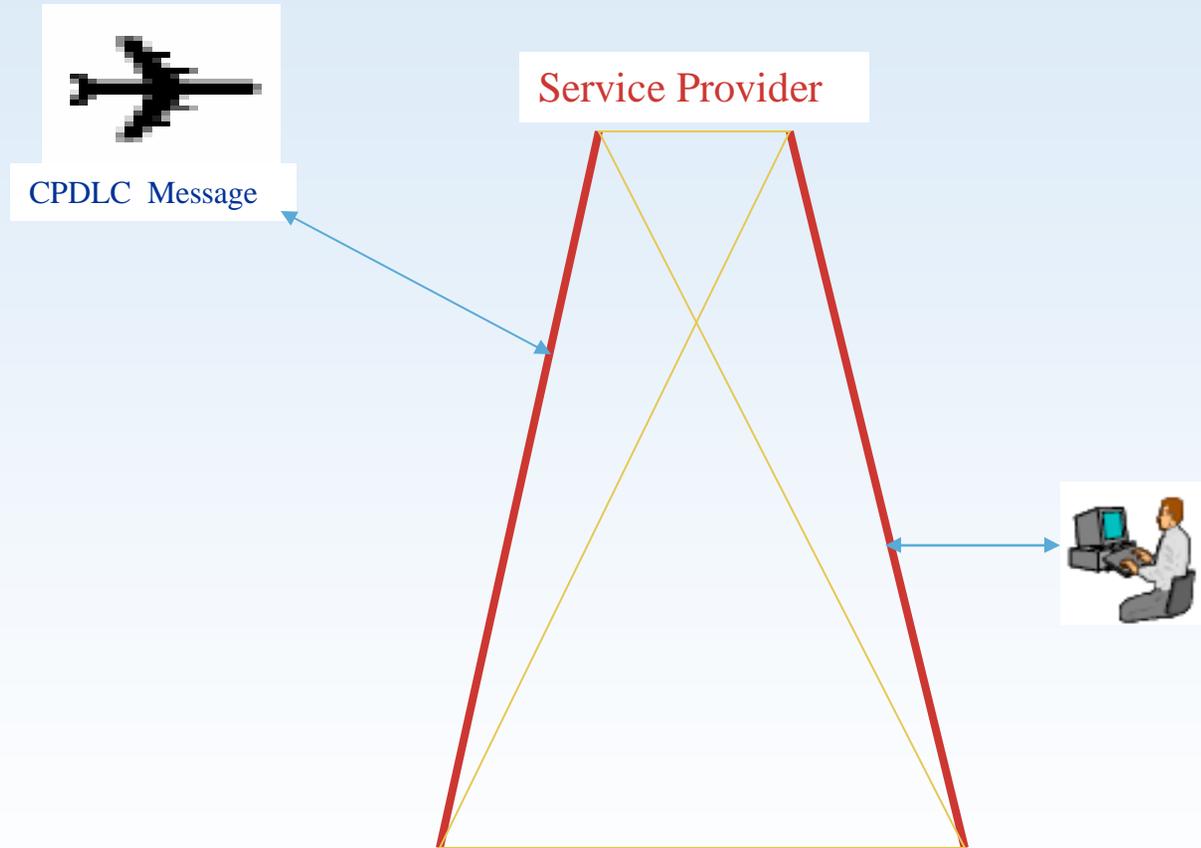
ATN Application Architecture [1/3]



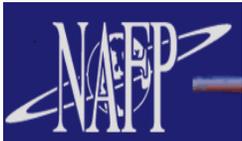
CM-logon Service



Downlink CPDLC Message (Air initiated)

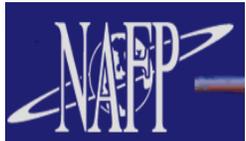


- **Security Threats**
 - Modification
 - Replay
 - Masquerade
- **Required protection**
 - Authentication:- verifying someone's identity
 - Data Integrity:- reassuring the recipient of a message that the message has not been altered since it was generated by a legitimate source
 - Confidentiality:-protect messages from unauthorized disclosure



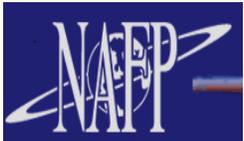
ATN Security Requirement [2/2]

- The Security Requirement 
 - End System Shall Support
 - ATN Key agreement scheme
 - ATN Digital signature scheme
 - ATN Message Authentication scheme



Elliptic Curve Cryptography

- ATN security provided by Elliptic Curve Public-Key Cryptography includes:
 - Elliptic Curve Digital Signature Scheme
 - Message Authentication Code Scheme
 - Key Agreement Scheme
- Implemented and combined properly, these schemes address the ATN Security threats

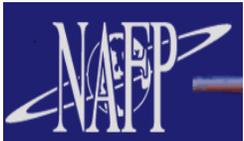


Elliptic Curve Domain Parameters

- The operation of each of these cryptographic schemes involves arithmetic operation on an elliptic curve over a finite field determined by:

1. Elliptic Curve Domain Parameters $T = (m, f(x), a, b, G, n)$

- T_{stan} :- used for key agreement and signing and verification by ATN application process
- T_{cer} :- used for certificate and certificate revocation lists signing by Certification Authority



Elliptic Curve Digital Signature Scheme

- Elliptic Curve Digital Signature Algorithm is described in terms of three primitives

- Key Generation Primitive (KGF)

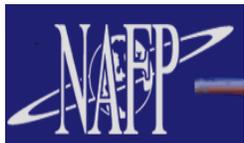
$$(d_U, Q_U) \leftarrow KGF_U(T)$$

- Signature Generation Primitive (EDSGA)

$$S_U \equiv (r, s) \leftarrow EDSGA(d'_U; M)$$

- Signature Verification Algorithm (EDSVA)

$$\tau \leftarrow EDSVA(M; S_U)$$



Elliptic Curve Message Authentication Code Scheme

- The MAC scheme is described in terms of three primitives:
 - Key Derivation Function (KDF)

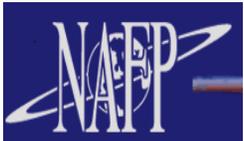
$$K_{U,V} \leftarrow \text{KDF}(Z_{U,V}; 80; \text{SharedInfo})$$

- The Tagging Algorithm (HMAC)

$$\text{MAC}_{U,V} \leftarrow \text{HMAC}(K_{U,V}; \text{SharedInfo})$$

- Tag Verification Algorithm

$$\tau \leftarrow \nu(K_{U,V}, \text{SharedInfo}, \text{MAC}_{U,V})$$



Public Key Infrastructure

- In the proposed protocol we assumed Certificate based PKI

Communicating Entity (U)

Generate (d_U, Q_U)

ID_U, Q_U

Certificate Authority (CA)

Generate (d_{CA}, Q_{CA})

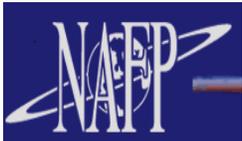
Q_{CA}

Generate: Cert (ID_U, Q_U)



Nomenclature

- Communicating Entities
 - Aircraft Entity Applications
 - ACM:- aircraft entity context management application
 - ACP:- aircraft entity CPDLC application
 - Ground Application Entity
 - GCM:- ground context management application entity
 - GCP:- ground CPDLC application entity
- Primitives
 - KGF:- key generation primitive
 - EDSA:- Elliptic curve digital signature primitive
 - SV:- secrete value derivation primitive
 - KDF:- key derivation primitive
 - HMAC:- message authentication code tagging primitive
 - MACV:- message authentication code verification primitive



Nomenclature *Conti.*

- Protocol Steps

- the form " $U \rightarrow V : [\text{info}]$ "

represents the communication of message, info, from U to V to ;

- The form $outdata \Leftarrow primitiveName(indata)$

represents the output (***outdata***) of a primitive (***primitiveName***) due to the given input (***indata***)

- Example

$$(d_U, Q_U) \Leftarrow KGF_U(T)$$

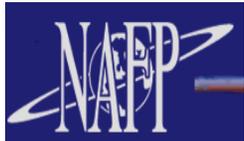
Shard Public Value Calculation

Aircraft entity CM Application

Ground CM application Entity

```
//Message creation and transmitting
Generate:
 $M_{req} \equiv (data, ID_{ACM}, ID_{GCM}, Time, ID_{GCP})$ 
Sign:
 $S_{ACM} \leftarrow EDSGA(d'_{ACM}; M_{req})$ 
Send:
" $ACM \xrightarrow{req} GCM[M_{req}, S_{ACM}]$ "
```

```
//Retrieval Process
From  $M_{req}$  get:
 $ID_{ACM}, ID_{GCM}, ID_{GCP},$  and  $Time$ 
Send:
" $GCM \xrightarrow{req} CA: [ID_{ACM}, ID_{GCM}, ID_{GCP}]$ "
Get:
" $CA \xrightarrow{resp} GCM: [CRL, Cert(ID_{ACM}, Q_{ACM}),$ 
 $Cert(ID_{ACM}, Q'_{ACM}), Cert(ID_{GCP}, Q_{GCP}),$ 
 $Cert(ID_{GCM}, Q_{GCM})]$ "
From the certificate get:,
 $Q_{ACM}, Q'_{ACM}$  and  $Q_{GCP}$ 
```



Shard Public Value Calculation

Aircraft entity CM Application

Ground CM application Entity

//Verification Process

Verify: the certificates using Q'_{CA}

Test: $Q_{ACM} \in CRL$ or $Q'_{ACM} \in CRL$ or $Q_{GCP} \in CRL$

If it is true stop

Verify: S_{ACM} using Q'_{ACM}

Verify: ID_{GCM} is its own ID

//Key Generation Process

Generate: a random value $Rand$

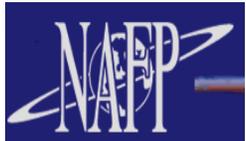
Calculate: $Z_{GCM,ACM} \leftarrow SV(d_{GCM}, Q_{ACM})$

Calculate:

$X_{GCM,ACM} \leftarrow KDF(Z_{GCM,ACM}; 80, 00_{16} || S_{ACM} || Rand)$

Calculate:

$K_{GCM,ACM} \leftarrow KDF(Z_{GCM,ACM}; 80, 01_{16} || X_{GCM,ACM} || cma || ID_{ACM} || ID_{GCM})$



Shard Public Value Calculation

Aircraft entity CM Application

Ground CM application Entity

//Message creation and transmitting

Generate:

$$M_{resp} \equiv (data, ID_{ACM}, ID_{GCM}, Rand, ID_{GCP}, Q_{GCP})$$

Sign:

$$S_{GCM} \leftarrow EDSA(d'_{GCM}; Time \parallel X_{GCM, ACM})$$

Calculate:

$$MAC_{GCM, ACM} \leftarrow HMAC(K_{GCM, ACM}; ID_{GCM} \parallel Count \parallel M_{resp} \parallel S_{ACM})$$

Send:

$$"GCM \xrightarrow{resp} ACM : [M_{resp}, MAC_{GCM, ACM}, Cert(ID_{GCM}, Q_{GCM})]"$$

//Retrieval Process

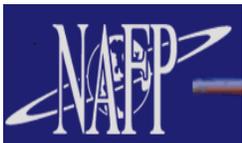
From M_{resp} get: $ID_{ACM}, ID_{GCM}, Q_{GCP},$ and $Rand$

From $Cert(ID_{GCM}, Q_{GCM})$ get: Q_{GCM}

//Verification Process

Verify: $Cert(ID_{GCM}, Q_{GCM})$ using Q'_{AC}

Verify: ID_{ACM} is its identity



Shard Public Value Calculation

Aircraft entity CM Application

Ground CM application Entity

//Key Generation Process

Calculate:

$$Z_{ACM,GCM} \leftarrow SV(d_{ACM}, Q_{GCM})$$

Calculate:

$$X_{ACM,GCM} \leftarrow KDF(Z_{ACM,GCM}; 80,00_{16} \parallel S_{ACM} \parallel Rand)$$

Calculate:

$$K_{ACM,GCM} \leftarrow KDF(Z_{ACM,GCM}; 80,01_{16} \parallel X_{ACM,GCM} \parallel cma \parallel ID_{ACM} \parallel ID_{GCM})$$

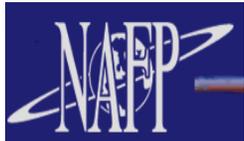


//Tag Verification Process

Calculate:

$$MAC_{ACM,GCM} \leftarrow HMAC(K_{ACM,GCM}; ID_{GCM} \parallel Count \parallel M_{resp} \parallel S_{ACM})$$

Test: if $MAC_{ACM,GCM} \equiv MAC_{GCM,ACM}$



Authentication Protocol

Key Agreement Protocol

Ground CPDLC application Entity

//Retrieval Process

From $Cert(ID_{ACM}, Q_{ACM})$ $Cert(ID_{GCM}, Q'_{GCM})$

Get: Q_{ACM}, Q'_{GCM}

From GCM entity gets: $X_{GCM, ACM}, Time, S_{GCM}$



//Verification Process

Verify: the Certificates using Q'_{CA}

Verify: the signature S_{GCM} using Q'_{GCM}



//Key Generation Process

Calculate: $Z_{GCP, ACP} \leftarrow SV(d_{GCP}, Q_{ACM})$

Calculate:

$K_{GCP, ACP} \leftarrow KDF(Z_{GCP, ACP}; 80; 01_{16} \parallel X_{GCM, ACM} \parallel cpd \parallel ID_{ACM} \parallel ID_{GCP})$



Aircraft entity CPDLS Application

//Retrieval Process

From the aircraft entity CMA get:

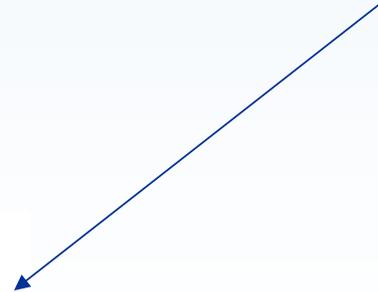
Get: $d_{ACM}, Q_{GCP}, X_{ACM, GCM}$



//Key Generation Process

Calculate: $Z_{ACP, GCP} \leftarrow SV(d_{ACM}, Q_{GCP})$

Calculate: $K_{ACP, GCP} \leftarrow KDF(Z_{ACP, GCP}; 80; 01_{16} \parallel X_{ACM, GCM} \parallel cpd \parallel ID_{ACM} \parallel ID_{GCP})$



Secret Key

$$K_{GCP, ACP} \equiv K_{ACP, GCP}$$



Authentication Protocol

Message Authentication For Uplink Messages

Ground CPDLC application Entity

//Message creation, tagging and transmitting

Generate: a message M_{GCP}

Calculate:

$$MAC_{GCP,ACP} \leftarrow HMAC(K_{GCP,ACP}; ID_{GCP} || count || M_{GCP})$$

Send: "GCP \longrightarrow ACP: [$M_{GCP}, MAC_{GCP,ACP}$]"

Aircraft Entity CPDLC Application

//Verification algorithm

Receive: M_{GCP} and $MAC_{GCP,ACP}$

Calculate:

$$MAC_{ACP,GCP} \leftarrow HMAC(K_{ACP,GCP}; ID_{GCP} || count || M_{GCP})$$

Verify: $MAC_{ACP,GCP} \equiv MAC_{GCP,ACP}$

Message Authentication For Downlink Messages

Ground CPDLC application Entity

//Verification algorithm

Receive: M_{ACP} and $MAC_{ACP,GCP}$

Calculate:

$$MAC_{GCP,ACP} \leftarrow HMAC(K_{GCP,ACP}; ID_{ACP} || count || M_{ACP})$$

Verify: $MAC_{GCP,ACP} \equiv MAC_{ACP,GCP}$

Aircraft Entity CPDLC Application

//Message creation, tagging and transmitting

Generate: a message M_{ACP}

Calculate:

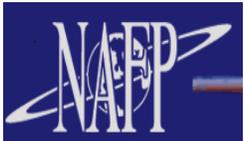
$$MAC_{ACP,GCP} \leftarrow HMAC(K_{ACP,GCP}; ID_{ACP} || count || M_{ACP})$$

Send: "ACP \longrightarrow GCP: [$M_{ACP}, MAC_{ACP,GCP}$]"



Conclusion

- *In conclusion, we hope the detailed authentication and key agreement protocol for CPDLC outlined in the paper may:*
 - *be used for guiding what assumptions and restriction should be imposed during the proposed simulation study*
 - *Serve as a reference during on going development of detailed specification on the protocol.*
 - *Facilitate the analysis of the ATN security solution*



QUESTIONS

