REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
18.Apr.02

3. REPORT TYPE AND DATES COVERED
THESIS

4. TITLE AND SUBTITLE
ENCRYPTED MODE SELECT ADS-B FOR TACTICAL MILITARY SITUATIONAL AWARENESS

5. FUNDING NUMBERS

6. AUTHOR(S)
2D LT JOCHUM JOHN R

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

8. PERFORMING ORGANIZATION REPORT NUMBER
CI02-41

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

20020523 150

14. SUBJECT TERMS

15. NUMBER OF PAGES
90

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT

ASC/PA
Bill Meers

Encrypted Mode Select ADS-B for Tactical Military Situational Awareness

by

2Lt. John R. Jochum

B.S., Electrical Engineering (1999)

United States Air Force Academy

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

Author...
Department of Electrical Engineering and Computer Science
April 27, 2001

Certified by..
Val M. Heinz
MIT Lincoln Laboratory
Thesis Supervisor

Certified by..
Dr. Thomas J. Goblick
MIT Lincoln Laboratory
Thesis Supervisor

Certified by..
Kenneth W. Saunders
MIT Lincoln Laboratory
Thesis Supervisor

Certified by..
Dr. Pratap N. Misra
MIT Lincoln Laboratory
Thesis Supervisor

Certified by...
Professor James K. Roberge
Department of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by...
Professor Arthur C. Smith
Chairman, Department Committee on Graduate Thesis

02-S-0812

THE VIEWS EXPRESSED IN THIS ARTICLE ARE THOSE OF THE AUTHOR AND DO NOT REFLECT THE OFFICIAL POLICY OR POSITION OF THE UNITED STATES, DEPARTMENT OF DEFENSE, OR THE U.S. GOVERNMENT

Encrypted Mode Select ADS-B for Tactical Military Situational Awareness

by

John R. Jochum

Submitted to the
Department of Electrical Engineering and Computer Science
on April 27, 2001,

in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Mid-air collisions between tactical military aircraft occur somewhat frequently and are a product of the inherent danger of dynamic maneuvering and flight at high airspeeds. Each collision results in significant loss to the military in terms of aircrew injury or fatality and airframe damage or loss. However, tactical aircraft do not have a collision avoidance or situational awareness aid. A potential solution to improve pilots' awareness of proximate traffic is Automatic Dependent Surveillance – Broadcast (ADS-B). Existing civil ADS-B systems are problematic for military use, because the broadcast data is not secure. In this thesis, we explore the feasibility of encryption for adapting civil Mode Select (Mode S) ADS-B to a secure military ADS-B system. Encryption theory applicable to ADS-B is reviewed. Modifications to the existing civil Mode S system required for encrypted ADS-B are identified, as well as required format modifications. The feasibility of encryption is discussed in terms of necessary security services for secure ADS-B: confidentiality, identification, authentication, and integrity. We examine the trade-offs between different types of conventional encryption techniques to provide these services, and conclude that encryption can be successfully applied to Mode S ADS-B. A feasible approach for encrypted ADS-B is presented.

Thesis Supervisor: Val M. Heinz
Title: Technical Staff, MIT Lincoln Laboratory

Thesis Supervisor: Kenneth W. Saunders
Title: Technical Staff, MIT Lincoln Laboratory

Thesis Supervisor: Dr. Thomas J. Goblick
Title: Technical Staff, MIT Lincoln Laboratory

Thesis Supervisor: Dr. Pratap N. Misra
Title: Senior Staff, MIT Lincoln Laboratory

Thesis Supervisor: James K. Roberge
Title: Professor of Electrical Engineering, Massachusetts Institute of Technology

Table of Contents

List of Figures

List of Tables

List of Acronyms

ADS-B	Automatic Dependent Surveillance-Broadcast
AES	Advanced Encryption Standard
ASIC	Application Specific Integrated Circuit
ATC	Air Traffic Control
ATCRBS	Air Traffic Control Radar Beacon System
CBC	Cipher Block Chaining Mode
CCI	Controlled Cryptographic Item
CDTI	Cockpit Display of Traffic Information
CFB	Cipher-Feedback Mode
CPLD	Complex Programmable Logic Devices
DES	Data Encryption Standard
DPSK	Differential Phase Shift Keying Modulation
ECB	Electronic Code Book Mode
EDC	Error Detection and Correction
GPS	Global Positioning System
HAE	Height Above Ellipsoid
ICAO	International Civil Aviation Organization
IFF	Identification Friend or Foe
IV	Initialization Vector
KVI	Key Validity Interval
LSB	Least Significant Bit
MAC	Message Authentication Code
MSB	Most Significant Bit
OFB	Output-Feedback Mode
OOK	On-Off Keying or Pulse Amplitude Modulation
PVT	Position, Velocity, and Time GPS estimate
SAC	Strict Avalanche Criteria
SSR	Secondary Surveillance Radar
TCAS	Traffic Alert and Collision Avoidance System
WGS 84	World Geodetic System, 1984

DF 17 fields:

DF	Downlink Format
CA	Transponder Capability
AA	Address Announced
ME	ADS Message
PI	Combined Parity and Interrogator Identity
II	Interrogator Identity

DF 19-specific fields:

AF	Application Field
MP	Message Payload
CF	Ciphertext Field

Acknowledgments

I would like to thank MIT Lincoln Laboratory Group 92 for sponsoring my thesis work for the last two years. From my initial visit to the Lab and throughout my stay at Lincoln, the professionalism and integrity you exhibited in engineering research taught me important lessons and made me proud to be a part of the Lab. I especially appreciate your efforts to connect my research with my professional interests and goals. The freedom to investigate various projects and the expertise and support of the technical staff made my research a challenging and enjoyable experience.

I would like to thank Professor James Roberge for his support and advice as my thesis advisor. I wish to thank Mr. Raymond LaFrey, the Associate Division Head of Division 9, for supporting my thesis work. I thank Dr. Steve Bussolari, Group Leader of Group 92, for the opportunity to work in Group 92. My thanks also goes to Dr. Clifford Weinstein, Group Leader of Group 62, for his time and insight.

Next, I would like to thank the United States Air Force Global Air Traffic Operations System Program Office and Air Combat Command for their support. I would especially like to thank Col. Stephen Henry, Col. Bud Vazquez, LtCol. Randy Nunley, LtCol. James Geib, Capt. Steve Hill, and Capt. Joe Slavinsky.

I would like to thank Dr. Zulficar Ramzan, of the MIT Cryptography and Information Security Group, for offering his time and recommendations for my thesis.

I wish to thank the Group 92 technical staff for all the time and effort they contributed to my thesis. I really appreciate all the help that Dr. Tom Goblick, Val Heinz, Dr. Pratap Misra, and Kenneth Saunders selflessly offered. Thank you for the technical insight, your reliable presence at thesis meetings, your guidance, and most of all for the lessons learned. You showed me the importance of a thorough and systematic approach to solving problems that I hope will prove invaluable down the road. I also wish to thank Group 92 staff that allowed me to infiltrate their projects: Barbara Chludzinski, Michael Hoffman, Dr. Ann-Marie Lind, Dr. Vince Orlando, and David Spencer. Finally, my thanks go to those in Group 92 and Group 18 who made sure everything ran smoothly from paychecks and computers to artwork for my thesis: Richard Lavoie, Vito Cavallo, John O'Rourke, Lynne Adamian, Pat Doiron, Robin Kennedy, Katharine Krozel, Cecilia Norris, Glenn Cook, Jennifer Leria, and Jennifer Weis.

1. Introduction

1.1 Problem Definition

Military aircraft and aircrews are lost for a variety of reasons, including controlled flight into terrain, collisions with other aircraft, and mechanical failure. From 1980 to 1997, Navy and Marine Corps aircraft suffered from 88 mid-air collisions, resulting in 134 fatalities and nearly $1.5 billion in aircraft damage [1]. Almost all of these collisions involved other military aircraft. Tactical aircraft suffered the majority of the collisions, due to the dynamic nature of flight at high airspeeds and formation flight at small separation distances. Collisions between military and civilian aircraft were extremely rare. When similar information from Army and Air Force mid-air collisions is considered, the resulting losses are much higher.

Automatic Dependent Surveillance – Broadcast (ADS-B) is a civilian system developed to provide situational awareness (SA) for pilots, where each aircraft broadcasts aircraft state parameters such as position and velocity. The broadcast data is received and displayed in nearby aircraft, which improves pilots' awareness of nearby air traffic. ADS-B systems can be used to alert the pilots to unsafe situations so that corrective action can be taken.

ADS-B offers a potential solution to military mid-air collisions, but civil ADS-B as it currently exists is problematic for military use. The primary difficulty is that data is broadcast in the clear. An enemy could intercept and exploit the ADS-B data for intelligence purposes. In addition, an enemy may "spoof" or create deceptive broadcasts. With existing ADS-B platforms, the user cannot differentiate an authentic broadcast from a spoofed broadcast. For tactical military applications, data must be transmitted securely between aircraft, and the use of encryption should be studied for secure ADS-B. The goal of this thesis is to explore the application of encryption to an air-to-air data link to help prevent military mid-air collisions through *secure* surveillance.

1.2 Air Traffic Control Technology

Modern air traffic control (ATC) is based on the use of radar to track aircraft. Information on aircraft location is displayed to ground controllers, who are responsible for issuing commands to maintain aircraft separation. Two types of radar are used to track aircraft. Primary radar tracks aircraft using reflected energy and does not determine aircraft altitude unless the radar is sophisticated and expensive. To supplement primary radar, ATC uses secondary surveillance radar (SSR) to digitally interrogate aircraft transponders at 1030 MHz, which respond with aircraft identification and altitude on 1090 MHz [2].

There are two primary types of civil transponders. The original transponder design is referred to as an ATCRBS (air traffic control radar beacon system) transponder. ATCRBS interrogations generate two replies, Mode A or Mode C, via pulse amplitude modulation or on-off keying (OOK), as shown in Figure 1-1. Mode A replies contain one of 4096 (12-bit) identification codes that are pilot-selectable on a transponder control panel. Mode C replies contain the aircraft's barometric altitude.

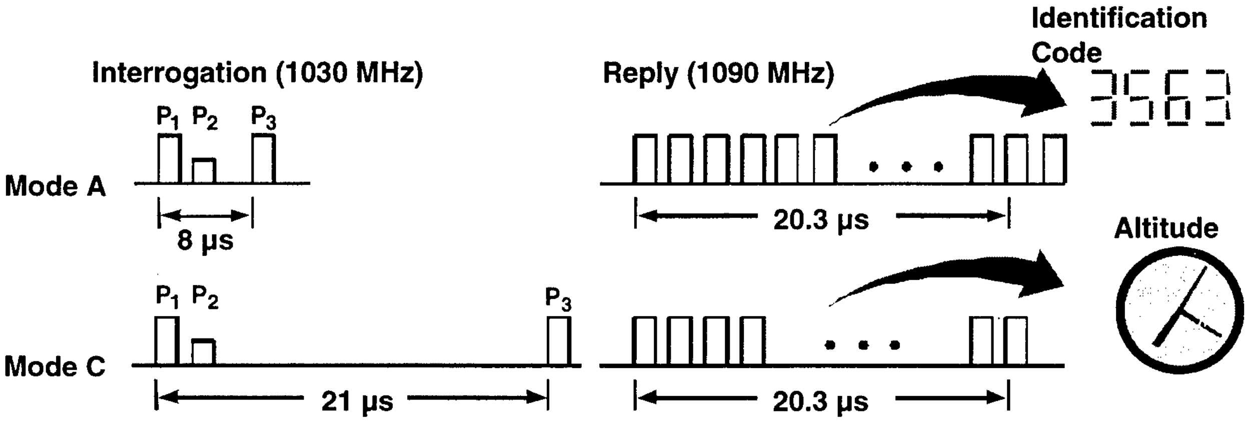


Figure 1-1. ATCRBS Mode A and Mode C interrogations and replies.

Mode Select (Mode S) is the second type of transponder, and it has added capabilities to reduce the limitations of Modes A and C. Mode S interrogations use differential phase shift keying (DPSK) modulation at a 4 Mbps rate, allowing 56 or 112-bit interrogations. Mode S replies use pulse position modulation (PPM) at a one Mbps rate, allowing for 56 or 112-bit replies. Each Mode S transponder is assigned a 24-bit hardware address that is unique worldwide, in addition to the 12-bit selectable ID for Mode A. Mode S transponders reply to interrogations with their 24-bit address and altitude. Advanced versions of this transponder also support two-way data link [3].

With the datalink feature, Mode S transponders are an enabling technology for automatic dependent surveillance (ADS), in which aircraft provide real-time broadcasts (ADS-B) of their own position in an omni-directional fashion [4]. To transmit this information in the open, Mode S transponders spontaneously "squitter" or broadcast a digital 56 or 112-bit message without being interrogated. As shown in Figure 1-2, the 56-bit short squitter contains the 24 bit Mode S address plus control and parity bits, and is broadcast once per second [4]. The 112-bit broadcast, known as an extended squitter, may be broadcast several times a second [4]. The additional 56 bits in the ADS message field can contain aircraft state data, such as position and velocity.

(8)	(24)	(24)
Control	Mode S address	Parity

DF 11 Short Squitter, 56 bits at 1 Mbps

(8)	(24)	(56)	(24)
Control	Mode S address	ADS Message	Parity

DF 17 Extended Squitter, 112 bits at 1 Mbps

Figure 1-2. Mode S ADS-B squitter format for the short and extended squitters. The number of bits for each field are shown above the field in parentheses. The short squitter message format is known as Downlink Format (DF) 11, while the extended squitter message format is DF 17.

During the 1980s, the civil air transport community developed and adopted TCAS technology to reduce the chances of mid-air collisions. TCAS, which stands for traffic alert and collision avoidance system, employs two avionics units to track nearby aircraft: the transponder and the TCAS processor and antenna package.

TCAS helps prevent collisions by interrogating Mode A/C aircraft and Mode S aircraft in different modes, as shown in Figure 1-3 [5]. TCAS interrogates ATCRBS-equipped aircraft with the "whisper-shout" technique, in which the control segment of the digital interrogation contains pulses at variable power levels to selectively acquire Mode C altitude from aircraft at different ranges. This Mode A/C interrogation disables replies from Mode S equipped aircraft. Interrogation of Mode S equipped aircraft uses passive "squitter listening" to first intercept the short squitter of other Mode S equipped aircraft, thereby acquiring the 24-bit address of nearby aircraft.

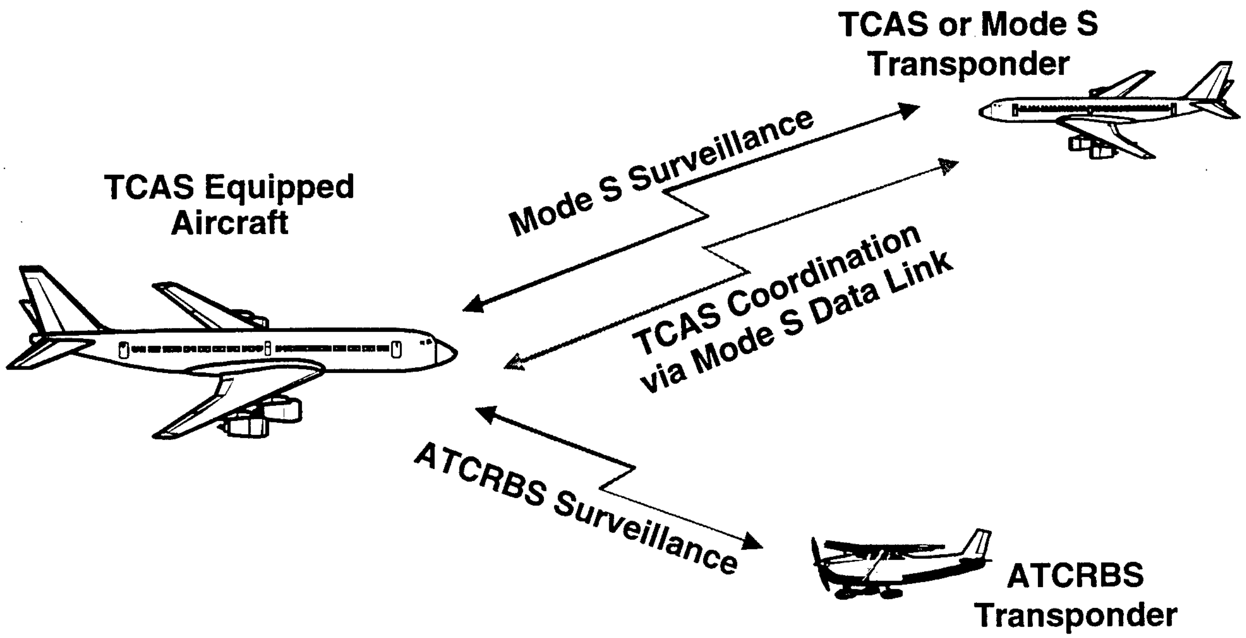


Figure 1-3. TCAS use of Mode S and ATCRBS.

Once the addresses are acquired, TCAS discretely interrogates these aircraft on the 1030 MHz frequency as if it were a ground-based Mode S SSR. The TCAS unit receives and decodes replies at 1090 MHz with the aid of a directional antenna, which determines relative bearing to nearby aircraft. The round trip signal travel time determines range, and the Mode C/S replies contain altitude information. Mode S replies also contain the Mode S address to aid in tracking. Aircraft are displayed to the pilot on a small screen called a cockpit display of traffic information, or CDTI. Aircraft that pass certain range, range-rate, and altitude tests with respect to the TCAS-equipped aircraft are highlighted to indicate potential for collision, and these warnings are referred to as traffic advisories. A more advanced form of TCAS, called TCAS II, can also issue resolution advisories to pilots advising them to climb or descend to avoid a collision.

1.3 Military Situational Awareness

US military aircraft often fly in formation to maintain unit integrity and mutual fire support in combat scenarios [6]. Fighter aircraft also fly in formation to join up for in-flight refueling and to promote decreased radar signature during ingress and egress of the target area. These aircraft may converge to within 30 feet to perform in-flight aircraft inspections [7].

Moreover, formation flight sometimes occurs at night and with poor visibility, which can lead to high pilot workload and loss of situational awareness. Fighter pilots also regularly train for air combat maneuvering (ACM) or dog-fighting, which requires very dynamic flight maneuvers and high g-force loading. It is not uncommon for aircraft to lose sight of other fighter aircraft, which can lead to unexpectedly high closure rates.

As these scenarios illustrate, tactical military aircraft have much higher performance than civilian aircraft, with less predictable flight paths and higher closure rates. As opposed to the civil ATC practice of ensuring large separation during flight, military training and combat doctrine require aircraft to converge. Moreover, most tactical aircraft cockpits are filled with equipment, and it is difficult and very costly to introduce new avionics. Because of these complexities and requirements, TCAS is not an option for tactical aircraft collision avoidance. TCAS is not adaptable to tactical military scenarios, and installation of new avionics and two new antennas is undesirable.[1]

An alternative with considerable flexibility is a Mode S ADS-B approach to military situational awareness. Mode S transponders can broadcast squitters using their existing 1090 MHz transmitters. To receive squitters, military aircraft would have to equip with 1090 MHz receivers, which could share existing transponder antennas. ADS-B processor cards could be added to transponders, which would result in a negligible impact on overcrowded avionics bays. The primary disadvantage of a civil Mode S ADS-B approach is that the broadcasts are not secure.

This thesis examines how to adapt civil Mode S ADS-B for military use. Civilian ADS-B measurements conducted by Lincoln Laboratory in the United States and Europe indicate that extended squitter messages can be received by ground stations located over 100 miles away [8, 9]. In combat scenarios, enemy forces could use this information to anticipate attacks and prepare defensive measures. During peacetime, enemy agents could intercept and record ADS-B messages from tactical aircraft at training ranges and use this information to derive U.S. military tactics and capabilities.

[1] A modified version of TCAS has been successfully implemented on a number of heavy military transport and tanker aircraft.

Two techniques can be used to overcome Mode S security problems. The first technique requires aircraft to broadcast squitters at reduced power levels, making them harder to receive at long range. The second technique involves encrypting the squitters so that interception of the squitters by unauthorized parties does not yield useful information. This thesis deals only with encryption alternatives for Mode S squitters.

2. Objective and Approach

2.1 Thesis Objectives

The objective of this thesis is to determine the feasibility of encrypting a Mode S ADS-B air-to-air datalink. The motivation for the use of encryption is to secure the datalink, thus providing heightened situational awareness to pilots of tactical military aircraft without the fear of system corruption or exploitation by adversaries. An increase in situational awareness through ADS-B should help prevent mid-air collisions for tactical aircraft.

In order to provide a secure communication link, a cryptographic system should provide several security services [10]. Confidentiality, identification, authentication, and integrity are services that are particularly necessary for the military ADS-B system. Confidentiality ensures that the information transmitted is accessible only to authorized parties. Identification enables own aircraft to identify the sender of all received messages through a discrete address. Authentication ensures that the identity of a received message is not false and that the message was generated by an authorized user. Integrity ensures that transmitted information has not been modified before or during transmission.

The feasibility of an ADS-B encryption system will be addressed in this thesis by determining viable methods to provide these security services. From this primary objective, several subordinate objectives also arise:

- The encryption system should provide desired security services and be strong enough to incorporate a margin of safety that exceeds adversaries' capabilities to attack the system, rendering all attacks infeasible. The system should guarantee the security of the information during the tactical time frame that the information is useful.
- The encryption system should be secure for the lifetime of the system, based on anticipated growth of processing capability and adversary assets.
- Although an encrypted squitter would be used solely for military surveillance, the encrypted squitter will be used in conjunction with other Mode S services. Therefore, an encrypted Mode S ADS-B system should integrate in a straightforward manner with existing services. Message content as well as error detection and correction capability should not degrade due to the application of encryption. The application of

encryption should allow squittered information to be compatible with other civil ADS-B squitters.

- An encrypted squitter may be developed as a subtype format for a military squitter. The use of encryption should permit flexibility, so that other military applications may utilize the squitter for both encrypted and non-encrypted applications.

2.2 Approach

The remainder of the thesis is organized as follows: Chapter 3 introduces encryption theory that is applicable to ADS-B. It discusses the conventional methods used to provide security services by looking at the elements of encryption systems, to include algorithms and keys. It also takes a closer look at the ADS-B environment and the types of attacks applicable to ADS-B.

Chapter 4 elaborates upon the ADS-B system required to generate and receive Mode S extended squitters. The chapter first describes the hardware architecture for a military ADS-B system, and discusses additional subsystems required for encryption. Next it introduces formatting concerns through analysis of the existing civil extended squitter. The chapter also provides an overview of Mode S error detection and correction and specifically focuses on the impact of encryption on error correction in a military ADS-B environment.

Chapter 5 discusses the feasibility of encryption. It does so by exploring conventional encryption methods to provide security services for an ADS-B extended squitter. The primary approach taken is to address specific vulnerabilities and attacks that may weaken a Mode S ADS-B encryption system. The vulnerabilities discussed are primarily patterns that arise due to formatting and content of the messages. Both passive and active attacks are also analyzed as potential concerns for the ADS-B encryption system. Methods to provide security services despite vulnerabilities and attacks are presented.

Chapter 6 analyzes the feasibility of encrypted ADS-B by weighing the methods provided in Chapter 5. Recommendations for encrypted ADS-B are provided, and a feasible approach is identified.

3. Encryption Theory Applicable to ADS-B

Encryption is a means to provide secure, confidential communication for the Mode S ADS-B extended squitter. Although other security measures such as steganography (hiding secret messages in larger messages without encryption) and power control exist, these alternatives are less secure than encryption or would not accommodate short Mode S ADS-B messages. This chapter reviews encryption theory as it applies to ADS-B.

The objective of encryption is to render information useless to all parties except those authorized to access it. The data to be transmitted securely, called the plaintext, must be disguised through an encryption process that generates the encrypted message or ciphertext. The ciphertext can then be sent securely to authorized recipients where it can be decrypted, or changed back into the original plaintext.

An encryption algorithm or cipher is the mathematical function used to encrypt the plaintext and decrypt the ciphertext. The fundamental building blocks of encryption algorithms are confusion and diffusion. Substitution is an example of confusion, and it creates non-linearities in the relationship between plaintext and ciphertext, making it more difficult for cryptanalysts to find redundancies or statistical patterns in the ciphertext [11]. Algorithms implement substitution through S-boxes, and S-box design is often critical to the strength of the algorithm. Diffusion is accomplished through permutations, in which redundancies in the plaintext are spread out and reordered in the ciphertext [11]. Algorithm objects called P-boxes permute the plaintext bits. Algorithms achieve security by performing multiple rounds of confusion and diffusion on the plaintext. For example, the Data Encryption Standard (DES) algorithm applies 16 rounds to the plaintext, and a single DES round is shown in Figure 3-1 [11].

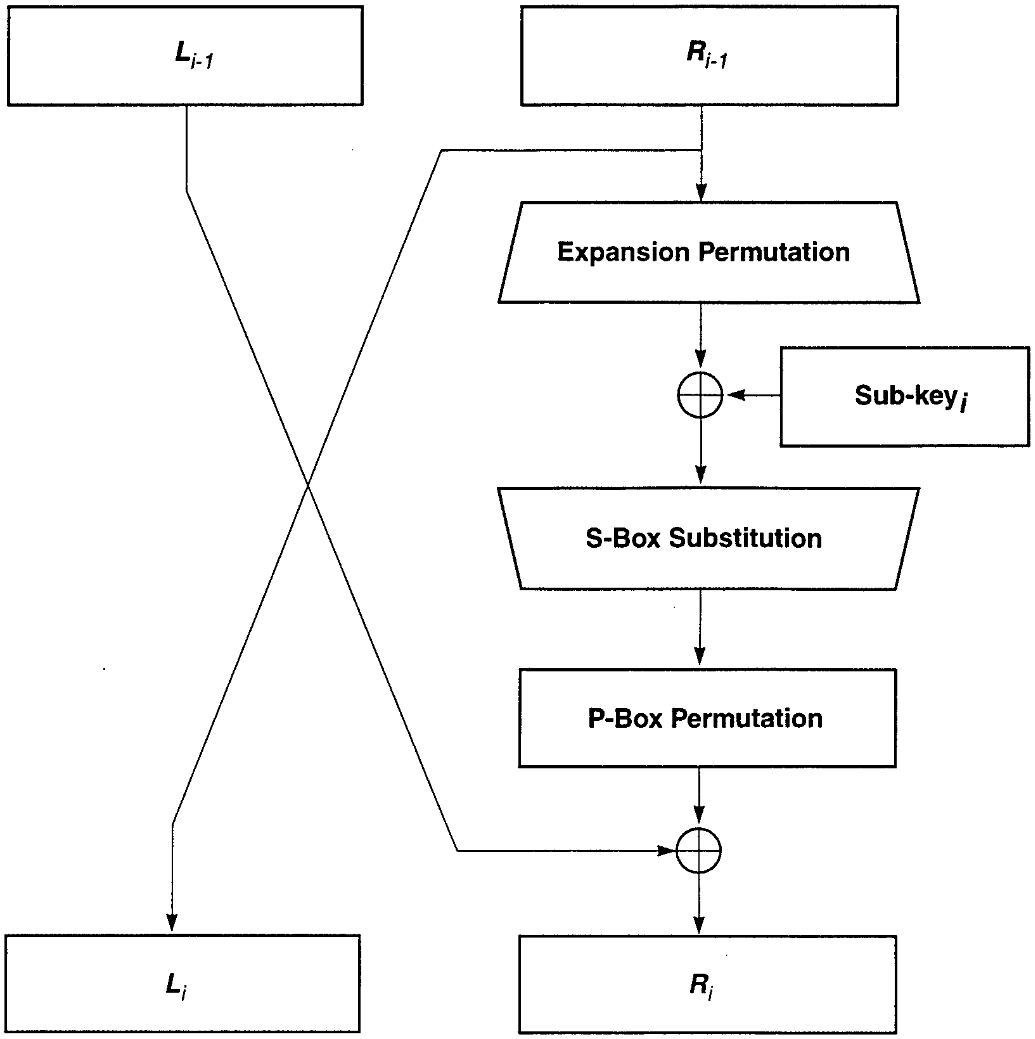


Figure 3-1. One round of the DES encryption algorithm. The round is iterated 16 times for a single plaintext block. As is common with Feistel networks, the input block to each round is divided in half, or 32-bit segments, as the data is passed through the algorithm. The right half of the input block (R_{i-1}) is expanded to 48 bits and then combined with a sub-key. The 48-bit result is divided into 6-bit segments that are passed through eight S-boxes, which condenses the block back to 32 bits. Next the block is permuted through a P-box. The result is then exclusive-ored with the left input block (L_{i-1}) before being passed to the next round. The input block R_{i-1} becomes the left input block L_i for the next round, without any additional computation.

Plaintext scrambling by the encryption algorithm depends upon a parameter known as a key. The key is a string of bits and should be chosen randomly. The key must also remain unknown to unauthorized users. A common assumption in cryptography is that the details of an algorithm are public knowledge, and that the security of an encryption system is based almost entirely in the strength and secrecy of the key [11]. This assumption fails for poorly-designed algorithms that allow shortcuts to cryptanalytic attacks. Therefore, it is essential to choose an algorithm that has no known weaknesses. Although it is not presently possible to guarantee a given algorithm is not vulnerable to shortcuts, choosing an algorithm that has been tested and proven extensively in the public cryptographic community is recommended.

3.1 Symmetric and Asymmetric Systems and Algorithms

Two types of cryptographic systems or cryptosystems can be considered for implementing a secure squitter. The systems are differentiated and defined by their keys. Symmetric cryptosystems rely on exchange of symmetric or secret keys which must be distributed in secret. The algorithms for these systems are called symmetric algorithms. In contrast, asymmetric protocols exchange asymmetric keys, and the users can exchange key information in public. One example of asymmetric algorithms are public-key algorithms, where the user provides a publicly-available key for anyone to access, and maintains a private key. The private key is computationally difficult to determine from the public key [11].

Asymmetric algorithms are attractive because they eliminate the need to exchange symmetric keys in secret. However, they are more complex and require greater keylength. In practice, asymmetric algorithms are used most often to encrypt and exchange symmetric keys as shown by Figure 3.1-1 [11]. Once accomplished, this allows more efficient exchange of data with a symmetric cryptosystem. Exchange of asymmetric key information, used for example by public-key encryption, prior to flight or in the air between aircraft is not presently feasible. This is largely due to limited bandwidth for the ADS-B air-to-air link, and also to the requirement to update keys frequently. Therefore, asymmetric cryptosystems and algorithms are not considered further in this thesis.

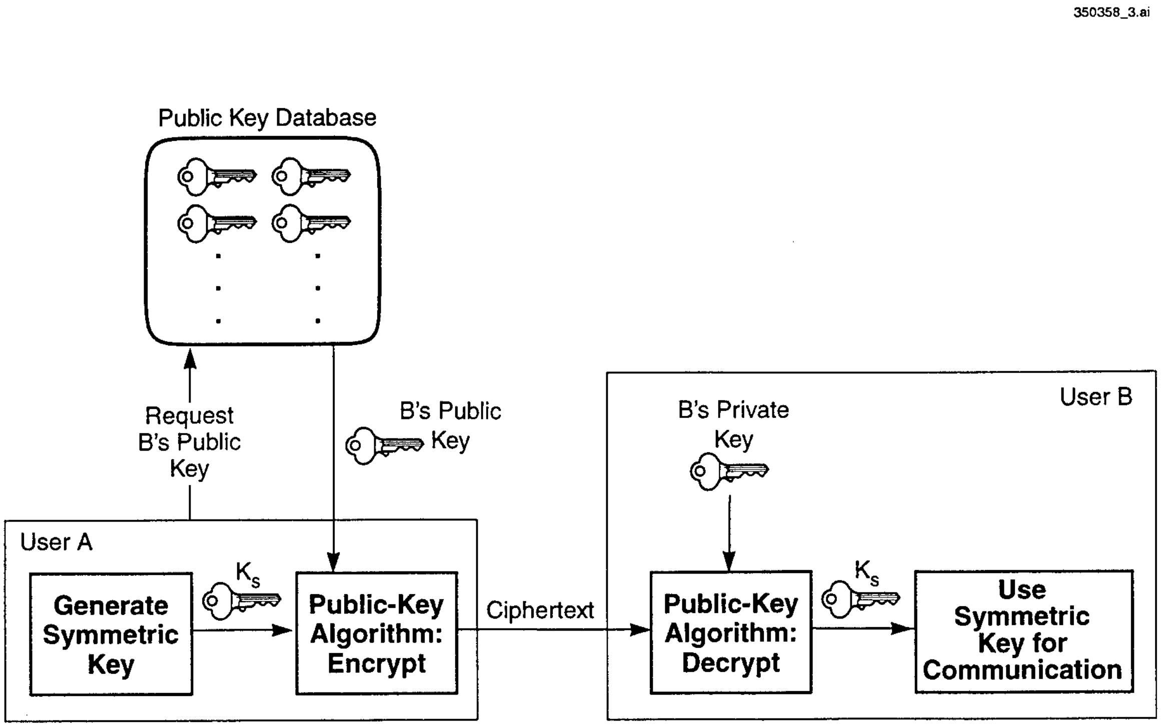


Figure 3.1-1. Use of a public-key cryptosystem to exchange symmetric keys. Two users, A and B, want to communicate with a symmetric cryptosystem. User A first obtains User B's non-secret public key by accessing a public database, or through direct communication that need not be secure. A then generates a symmetric key and encrypts it with the chosen public-key algorithm and B's public key. B then decrypts the message using his private key and the public-key algorithm. As a result, both have the symmetric key and can communicate using a symmetric cryptosystem [11].

For a symmetric cryptosystem, the symmetric key is used with the chosen algorithm to both encrypt and decrypt the message, as shown in Figure 3.1-2 [10]. If the key is compromised or needs to be updated frequently, all group members must be re-keyed with a new symmetric key, which may create timing, bandwidth, and storage difficulties.

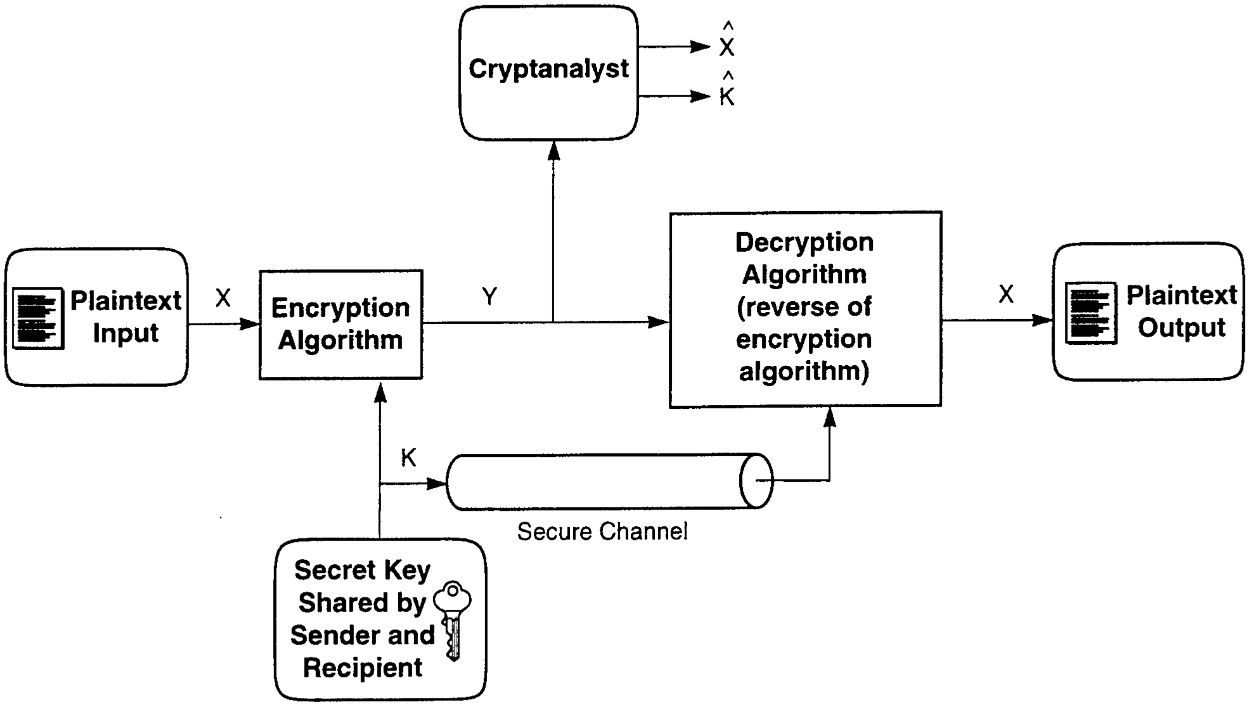


Figure 3.1-2. Symmetric key encryption. Both the sender and the recipient must first exchange the symmetric (secret) key through some secure means. To send a message, the sender encrypts the plaintext with the symmetric encryption algorithm and key, generating ciphertext. For ADS-B, the sender then broadcasts the ciphertext across an insecure channel, where a cryptanalyst may intercept the message and estimate the plaintext ($\hat{x}$) or the key ($\hat{k}$). Note the cryptanalyst cannot understand the ciphertext without the key. Next, the ciphertext arrives at the recipient where the identical symmetric algorithm is used in reverse order with the key to generate the plaintext.

3.2 Block and Stream Symmetric Algorithms

Symmetric algorithms are categorized into block and stream algorithm types. The primary difference between the two is the size of the plaintext information on which the algorithms operate. Block algorithms rely on the same function and a symmetric key to transform fairly large blocks of the plaintext, usually in multiples of 64 bits. Therefore, the same block of plaintext with a fixed key will always encrypt into the same ciphertext. Users typically implement block algorithms with some chaining or cryptographic feedback mode, which helps remove patterns generated due to similar or repeated plaintext. An advantage of block ciphers is their flexibility with a variety of modes. In addition, block algorithms are generally more suitable for software implementation because they operate efficiently on multiple bytes of plaintext [11].

The other category of symmetric algorithms is stream algorithms, which typically encrypt smaller pieces of the plaintext, such as one bit or one byte at a time. Stream algorithms are also generally faster than block ciphers, especially when implemented in hardware, because bit-wise operations can be implemented efficiently in silicon [11]. A very basic stream implementation is called a keystream generator, and it consists of three standard parts: the internal state, a next-state function, and an output function, as shown in Figure 3.2 [11]. The internal state describes the algorithm's current state. The next-state function updates the internal state based on the symmetric key K and contains the cryptographic complexity of the algorithm. The output function generates a single keystream bit K_i based on some simple operation on the internal state. Although the keystream is deterministic and periodic, out of necessity the keystream period is much longer than the length of the plaintext to be encrypted. If the keystream does not repeat during a given interval, despite a fixed symmetric key, two identical plaintext inputs separated in time by segments of the keystream produce different ciphertext. Note that stream algorithms require keystream synchronization between the sender and recipient. One disadvantage of stream algorithms is that they do not implement diffusion and rearrange bit dependencies if the algorithm does not include some type of ciphertext feedback.

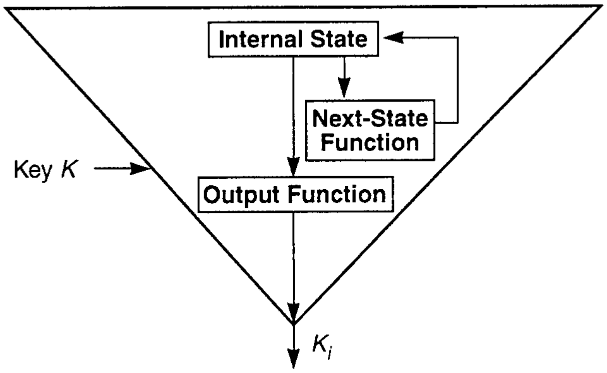


Figure 3.2. Stream algorithm block diagram when used as a keystream generator. A symmetric key K is passed as a parameter to the algorithm, and a stream of bits, or keystream K_i, is generated by the algorithm. The keystream then combines with the plaintext through some operation such as a simple exclusive-or to generate ciphertext. Both the sender and the recipient must synchronize keystreams.

3.3 Security and the ADS-B Environment

The expected security of strong, proven symmetric algorithms is based largely on the strength of the key. If there are no inherent weaknesses in the algorithm that can be exploited, the assumption made is that the only means to compromise the cryptosystem is to test every key in the key space through a brute-force attack [11]. The strength of the key then becomes proportional to the number of bits within the key, or the key length. The complexity of a brute-force attack, which is one means to measure the security of the cryptosystem, is related to the time required to complete the attack. This is determined by

$$t_{brute-force} = \frac{2^n}{r} \tag{3.3-1}$$

where n is the number of bits in the key and r is the key-test rate in number of keys per unit time. If there are 2^n possible keys, the system will be compromised with a 50 percent chance after testing half of the keys.

The security of the system and the requisite key length are largely determined by external factors, such as the processing capabilities and anticipated resources of the adversary. For the purposes of encryption, an adversary is any unauthorized party attempting to intercept or interfere with confidential broadcasts. Prudent system design calls for keylength that exceeds by a significant safety factor the projected processing capabilities of an adversary, based upon the value of the information to be encrypted [11]. In other words, the value of military ADS-B information must be estimated, and encryption must protect that information based on assets an adversary is willing and able to utilize. Moore's Law, which states that processing speed divided by price doubles every 18 months, will be used as a rule of thumb to project adversary assets and the required keylength.

Additional aspects of an ADS-B cryptosystem are the environment in which it will operate and the types of attacks it must protect against. Figure 3.3 describes the type of security attacks applicable to the ADS-B environment [10].

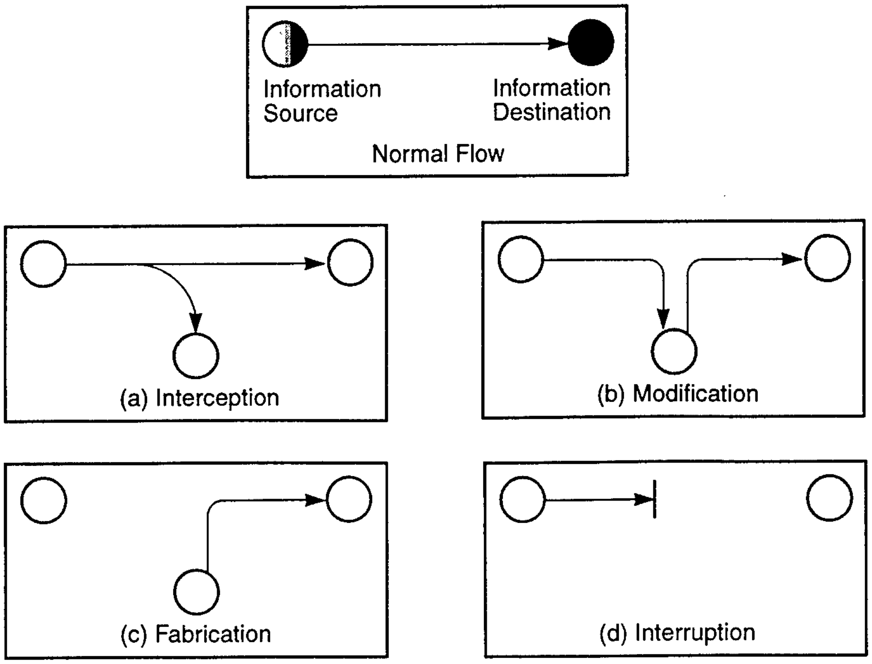


Figure 3.3. ADS-B security threats.

The first attack, interception, is an attack on confidentiality. In the ADS-B air-to-air environment, an adversary can intercept a broadcast and attempt to compromise the key. Protection against this attack relies on the strength of the algorithm and key. The second type of attack is modification, where an adversary tampers with the squitter, attacking its integrity. Authentication can help combat this attack, by recognizing and discarding altered messages. Fabrication is a second type of attack on authenticity, where an adversary inserts counterfeit squitters into the system. Again, authentication and identification can prevent fabrication attacks. The final attack applicable to ADS-B is interruption, which attacks the availability of squitters by jamming squitter transmissions. There is not a cryptographic solution to this attack, and it will not be further addressed.

4. Mode S Extended Squitter

This chapter describes the Mode S ADS-B system required to broadcast and receive extended squitters. It describes additional functionality and modifications to existing hardware needed for cryptographic capability. The chapter also looks more closely at the details of Mode S squittering to lay the foundation for analysis of encryption methods in subsequent chapters.

Section 4.1 provides an overview of a military ADS-B system. The section discusses subsystems within the aircraft transponder required for encrypted squittering, as well as required interfaces with subsystems external to the transponder such as navigation systems and displays. Emphasis is placed on novel subsystems required for encryption. The section also discusses the steps required to communicate with encrypted squitters.

Section 4.2 describes the civil extended squitter, known as Downlink Format (DF) 17. This section introduces formatting concerns by providing a timing diagram for generating position squitters. The DF 17 format will also serve as a base model for the encrypted squitter format in subsequent chapters.

Mode S error detection and correction is described in Section 4.3. Again, the emphasis is placed on the impact of encryption on existing error detection and correction functionality, if any. The section discusses the error environment for military ADS-B and discusses modifications necessary for encrypted squitters.

4.1 Military Mode S System Description

Encrypted Mode S ADS-B for military situational awareness can utilize Mode S / IFF transponders. System implementation will require additional functionality, including an ADS-B processor, a 1090 MHz squitter receiver, a timing subsystem, and a crypto-unit to encrypt and decrypt squitters. Figure 4.1-1 illustrates the various subsystems required for encrypted squittering, and this section will describe the functions and relationships of these subsystems [12].

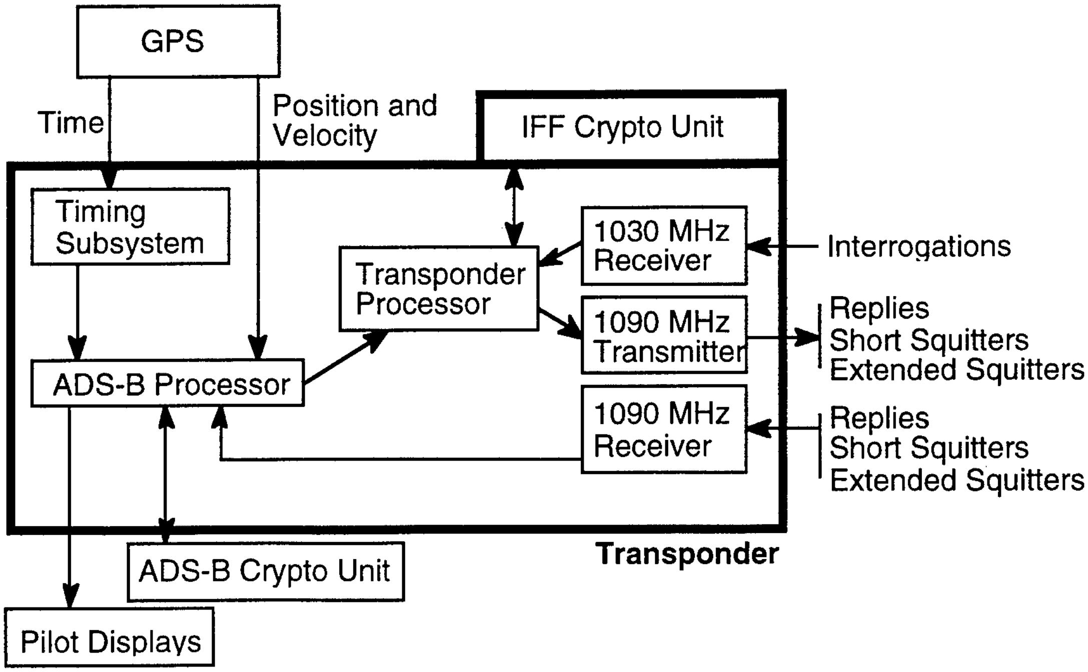


Figure 4.1-1. Military Mode S ADS-B system description. Several subsystems interface to enable identification of friend or foe (IFF), encrypted Mode S ADS-B, and compatibility with civil ATC systems. The ADS-B processor, 1090 MHz receiver, and ADS-B crypto-unit are required for encrypted squittering.

Several of the subsystems necessary for ADS-B are already built into the transponder. For example, the 1090 MHz transmitter and antennas will be used to broadcast extended squitters. The transponder processor will also be used to extract data from transponder registers and parity encode squitters prior to broadcast. Existing navigation equipment such as GPS and inertial navigation systems will be interfaced, and existing displays should be modified to accommodate ADS-B information.

An encrypted ADS-B system will require novel subsystems. To receive extended squitters, a 1090 MHz receiver must be added to the Mode S / IFF transponder. The receiver will detect incoming Mode S reply formats, demodulate and digitize incoming squitters, and perform Mode S error detection and correction before passing data to the ADS-B processor. A timing subsystem is needed to accurately compare GPS position data received from other aircraft with own aircraft's position through squitter timing synchronization. The ADS-B processor and ADS-B crypto-unit are also additional subsystems required for encrypted ADS-B. The ADS-B processor serves several purposes:

- Own aircraft data:
  a) Accept data from own aircraft navigation systems
  b) Format own aircraft data to be transmitted
  c) If encrypted squitters are enabled, interface aircraft data with the crypto unit to encrypt own aircraft squitter data
  d) Upload data to transponder registers for outgoing squitters
- Data from other aircraft:
  a) Accept data from other aircraft (incoming error-corrected squitters)
  b) Interface with the crypto-unit to decrypt incoming squitters
  c) Decode data and determine relative position of other aircraft with respect to own aircraft
  d) Track the relative positions of other aircraft
  e) Calculate the trajectories of other aircraft relative to own aircraft
  f) Generate traffic data for the interface with a cockpit display
  g) Generate traffic alerts if traffic scenario meets alert criteria

The ADS-B crypto-unit stores key information, updates the symmetric keys, and encrypts/decrypts squitter plaintext for secure communication.

The IFF and ADS-B crypto-units are indicated by separate blocks outside the transponder block to distinguish them as controlled cryptographic items (CCI). This illustrates that the secret hardware and software for each system is centralized and segregated from other transponder subsystems. Essentially, the encryption capability for Mode S ADS-B should be easily removed from the transponder. This serves two purposes. First, the transponder remains fully operational should the encryption hardware need to be removed for repair or upgrade. Second, the secure information may be safeguarded more easily if it is a separate unit from the transponder. That is, a collection of plug-in cards that contain secret encryption keys may be secured in a single safe, as opposed to a requirement for armed security police to stand guard and secure transponders inside aircraft.

The transponder must also interface with navigation subsystems for Mode S ADS-B. The GPS unit interfaces with the ADS-B processor and is the primary data source for own aircraft position, velocity, and time (PVT). The GPS unit typically provides PVT updates each GPS second based on GPS time that is accurate to 120 ns [13]. Each PVT update is also accompanied by a time mark pulse, which synchronizes the transponder timing subsystem to GPS time.

The transponder timing subsystem divides the GPS second into 200 ms intervals with time ticks inside the transponder. Using these ticks, the ADS-B processor estimates own aircraft position using the PVT velocity estimate to extrapolate the aircraft position corresponding to each 200 ms tick. This allows multiple accurate position squitters per second, rather than redundant squitters. The timing subsystem also serves as a time tag for synchronized squitters. This enables aircraft to determine the time of validity for an incoming squitter, as squitter message content does not include time data. This will be discussed in greater detail in Section 4.2.

The following lists summarizes the subsystem interaction required to generate and broadcast an encrypted squitter, and to receive and decode an encrypted squitter:

Generate and broadcast encrypted squitters:	**Receive and decode encrypted squitters:**
• GPS unit generates PVT estimate and time stamp pulse on the GPS second • ADS-B processor receives and processes PVT estimate, generating plaintext • ADS-B processor extrapolates position each 200 ms time tick • ADS-B crypto unit encrypts ADS-B data • Transponder processor receives and loads ciphertext into registers • Transponder processor formats and parity encodes the squitter when a broadcast is triggered • 1090 MHz transmitter modulates and broadcasts the squitter	• 1090 MHz receiver detects, de-modulates, and digitizes received squitter • 1090 MHz receiver performs error detection and correction (EDC) • ADS-B processor receives squitter and recognizes encrypted format • ADS-B crypto-unit decrypts the ciphertext • ADS-B processor processes plaintext data • ADS-B processor extrapolates received squitter PVT data to own aircraft current time • Cockpit display provides information to pilot

The next section illustrates this process without encryption by taking a closer look at formatting and synchronization for the civil Mode S extended squitter. The process without encryption serves as a basis for the encrypted squitter. Section 4.3 will introduce Mode S error detection and correction, and also discuss the impact of encryption on this process.

4.2 DF 17 Extended Squitter

Although designed as an upgrade to secondary surveillance radar, Mode S transponders are also a vehicle for ADS-B through datalink with the extended squitter. As illustrated by Figure 4.2-1, the role of the squitter broadcast is to provide aircraft-derived position, identification, and state information for a variety of civil and commercial surveillance applications. To enable compatibility for multiple applications and avionics hardware, the International Civil Aviation Organization (ICAO) standardizes formats for Mode S messages including squitters. The extended squitter is known as Downlink Format (DF) 17, a name derived by the assignment of 17 (decimal) to the five-bit Downlink Format field of the squitter. This section introduces the DF 17 standard and expands on characteristics of the squitter that will impact encryption.

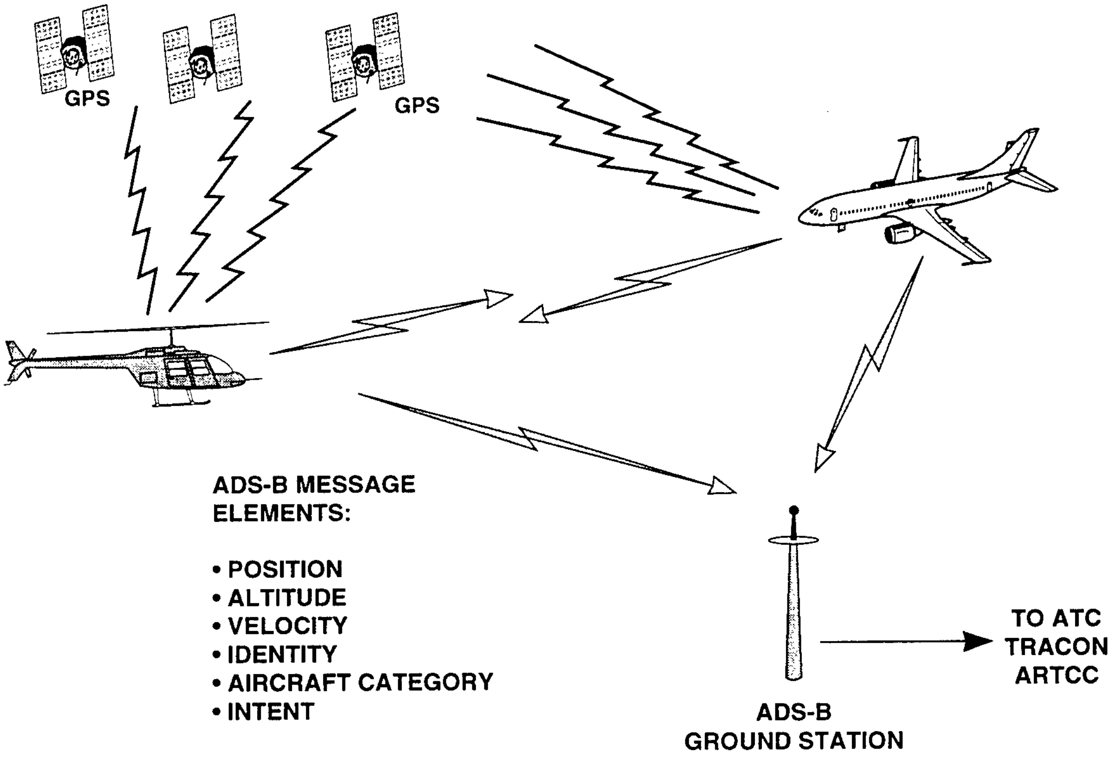


Figure 4.2-1. Mode S ADS-B concept with the DF 17 extended squitter. Navigation data is determined through GPS receivers that receive data from the GPS satellite constellation. Data is passed between aircraft through an air-to-air link. Air-to-ground data is also broadcast to ATC ground stations such as terminal radar approach control centers (TRACON) and air route traffic control centers (ARTCC).

As with downlink replies to ground radar interrogation, the DF 17 squitter signal is pulse position modulated with a carrier frequency of 1090 MHz and is broadcast at a one Mbps data rate [4]. The extended squitter is 112 bits long, as shown in Figure 4.2-2, and has five primary mission fields [14]. The Downlink Format (DF) is 5 bits long, identifies the squitter contents, and in this case is assigned decimal 17. The next field is the 3-bit transponder capability (CA) field, which describes the aircraft's Mode S performance capability. The address announced (AA) field identifies each aircraft's unique 24-bit Mode S address. The 56-bit extended squitter message (ME) field is essential for ADS-B, as it provides the message payload for aircraft parameters such as GPS position and velocity. Finally, the 24-bit parity (PI) field provides the parity for the extended squitter.

(5)	(3)	(24)	(56)	(24)
DF 17	CA	AA	ME	PI

Figure 4.2-2. Bit assignments and mission fields for the DF 17 ADS-B extended squitter.

Inside the ME field, the first 5 bits contain the format type code, which divides the DF 17 squitter into subtypes [15]. The squitter subtypes are provided in Appendix A and include identification, surface position, airborne position with barometric or GPS WGS-84 height above ellipsoid (HAE) altitude reporting, and airborne velocity subtypes. The airborne position types are further differentiated with respect to horizontal and vertical position errors. There are also subtypes for identification, aircraft intent, aircraft state, and testing.

The remaining data inside the ME field for each DF 17 subtype contains surveillance information as well as additional flags and fields specific to each subtype. Further information on the subtype ME fields can be found in the ICAO *Document 9688* [14]. Appendix A also illustrates examples of the ME field for DF 17 position and velocity squitters.

In addition to the DF 17 format, DF 17 processing and time synchronization are of particular interest for encrypted squittering. Time synchronization enables the ADS-B processor to determine time of validity for received squitter data which provides more accurate traffic displays for situational awareness. Figure 4.2-3 illustrates the timeline for synchronized DF 17 position squitters [12]. The timelines in this figure are the topics for the next several paragraphs.

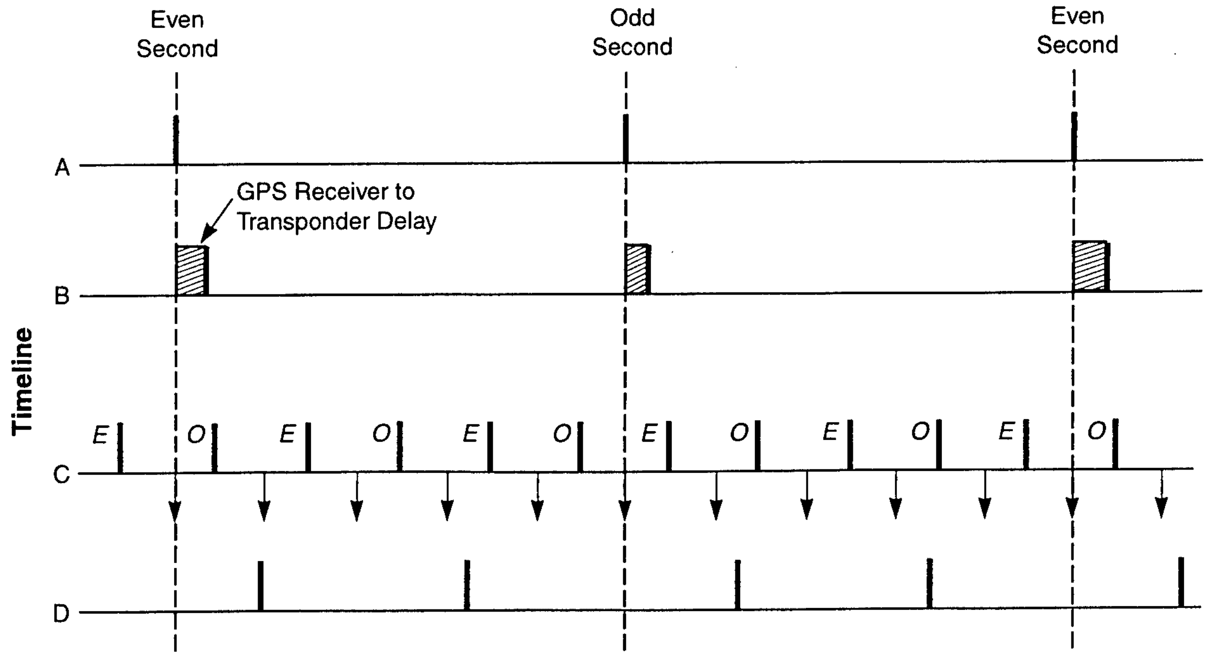


Figure 4.2-3. DF 17 extended squitter position update timelines, with the transponder timing subsystem synchronized to the GPS clock. Timeline A shows GPS position, velocity, time (PVT) updates each GPS second and the time stamp pulse sent to the transponder timing subsystem. Timeline B indicates upload latency from the GPS PVT data to the transponder ADS-B processor. Timeline C shows transponder internal 200 ms clock ticks, which enable compact position reporting and position extrapolation. Timeline D is the position squitter trigger, which prompts formatting, modulation, and broadcast of a squitter.

GPS with Precise Positioning Service (PPS) is the primary navigation source for military DF 17 ADS-B. System performance data used in this thesis is consistent with PPS. In addition, it is assumed that GPS data transfer occurs each GPS second, and that military aircraft transponders synchronize to GPS time during this transfer. A time synchronization bit inside the ME field indicates synchronization with GPS time. Synchronization occurs through an instantaneous time mark pulse sent by the GPS unit to the transponder timing subsystem, and the GPS time is accurate to 120 ns [13]. This enables updated GPS position, velocity, and time (PVT) estimates to upload to the ADS-B processor every second, as illustrated by Timeline A in Figure 4.2-3. Timeline B represents the time the PVT data actually arrives in the transponder registers, with a maximum latency of up to 200 ms [13].

Note that GPS PVT updates occur typically once each second. However, DF 17 position squitters occur on average twice a second, with a uniform distribution of 400 to 600 ms between squitters [14]. If an aircraft generates a second position squitter using the same GPS data generated nearly a second earlier, the data may provide a misleading representation of the aircraft's actual position. This is especially true for tactical military aircraft, where 600 knots corresponds to 1000 feet per second. To prevent redundant position squitters and to provide more accurate traffic displays, transponders estimate position for each 200 ms transponder tick by extrapolating position with the GPS velocity estimate. After extrapolation, the ADS-B processor loads the data into the transponder registers.

The synchronized 200 ms transponder ticks and register upload are shown by Timeline C in Figure 4.2-3. The arrows below Timeline C indicate the extrapolated position time of validity. Aircraft are able to determine the time of validity of received squitters by decoding the "E" and "O" Mode S formatting known as compact position reporting (CPR) [14]. CPR creates pseudo time-tags each squitter and condenses the GPS position bits to create an efficient and unambiguous position report.

The CPR time stamp is accomplished with the CPR format (F) bit, as illustrated by the ME field examples in Appendix A [14]. For synchronized squitters, the F bit toggles between even (E) and odd (O) on each transponder 200 ms update, as shown next to the ticks on Timeline C. An E tick corresponds to position extrapolated for an even 200 ms multiple after an even GPS second, and an O tick corresponds to position at an odd 200 ms multiple after the even GPS second. This creates position time of validity. For example, a position squitter received with an "E" F bit indication tags that squitter to the nearest transponder tick which occurred an even number of 200 ms time intervals after an even numbered GPS second. As such, the receiver can synchronize the position of an aircraft to a specific even or odd 200 ms epoch.

The second role of CPR is to reduce GPS position of 24 latitude and 24 longitude bits to an airborne report of 17 latitude and 17 longitude bits [14]. This provides efficient bit use in that high-order bits, which are very near constant, are not broadcast in every message. With position encoded in this manner, the F bit first enables globally unambiguous reporting by encoding the position data in two ways that differ slightly, where the differing reports are again designated E and O. Once the receiver decodes both an even and odd position report from the same aircraft, the ambiguous positions can then be superimposed to determine unambiguous position, providing approximately 17 m horizontal accuracy within the 95th percentile spherical error probable for a military PPS receiver, including CPR quantization error [16].

Timeline D indicates the squitter trigger. The triggers for each squitter subtype occur randomly. Once triggered, the squitter data is extracted from the transponder registers, formatted with control and address bits, parity encoded, modulated and broadcast. The next section discusses the Mode S parity encoding method, and analyzes the impact of encryption on Mode S error detection and correction.

4.3 Mode S Parity Encoding

Mode S transponders incorporate parity encoding to reduce both the probability of receiving incorrect messages and the need for re-transmissions. Parity encoding techniques were developed for Mode S so that aircraft transponders could perform interrogation error detection and ground sensors could perform reply error detection and correction (EDC) [17]. The discussion here will briefly cover the characteristics of downlink EDC, as the squitter is a form of downlink reply and is broadcast under the same considerations at 1090 MHz. This section will also analyze the impact of encryption on EDC, as error correction of squitters is absolutely required for encrypted ADS-B.

The Mode S downlink error environment at 1090 MHz is dominated by ATCRBS interference, in which Mode 3/A or Mode C replies may overlap the Mode S squitter. Multipath and TACAN signals are also interference sources. ATCRBS overlap, or synchronous garble, is more problematic than Mode S garble because ATCRBS ground radar requires 15 or more replies from each aircraft during a single sweep. In contrast, Mode S enables one interrogation / reply per scan for each aircraft, which greatly reduces synchronous garble [2]. Thus, Mode S parity encoding primarily overcomes errors due to interference, rather than random noise or signal fading [17]. ATCRBS interference is characterized by 0.45 μsec OOK pulses, separated by 1.00 μsec. As shown in Figure 4.3-1, ATCRBS framing pulses F_1 and F_2 are separated by 20.3 μsec [17]. Thus, a single ATCRBS reply would adversely affect a span of approximately 24 Mode S bits.

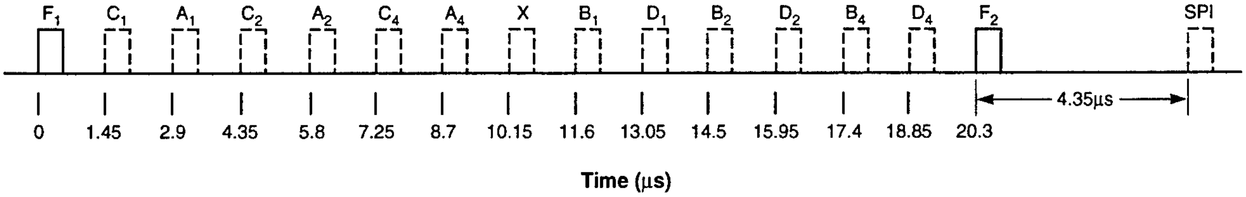


Figure 4.3-1. ATCRBS reply format, characterized by 20.3 μsec framing pulses. The reply creates a burst error that affects approximately 24 Mode S bits. The Special Position Indicator (SPI) pulse, activated by the "Ident" button on the transponder control panel, is not present for the majority of ATCRBS replies.

This error pattern is characterized by errors that tend to be bunched within a given span of bits, and the channel producing this pattern is called a burst error channel. Specifically for Mode S EDC, the burst error channel is dominated by ATCRBS interference of length equal to 24 Mode S bits, with errors likely to occur in the first, last, and any set of interior bits in the 24-bit span [17].

Burst errors will continue to be the most problematic error source in an encrypted squittering environment as well. In a military training scenario with proximate civil traffic, the ATCRBS interference would be dominant. In a more hostile environment or remote training environment with less ATCRBS interference, synchronous garble from overlapping squitters may become the principal error source. However, Mode S replies and squitters use PPM which allows greater bit interference detection by comparing energy received in each chip of individual data bits. This allows correction of overlapping replies by using the Mode S EDC burst parity code, so long as the replies overlap less than 24 bits. In addition, Mode S squitters have lower repetition frequencies than ATCRBS interference, making them less problematic [2]. If multiple ATCRBS signals garble a received squitter over a span greater than 24 bits, or if squitters overlap more than 24 bits, the error is detected and declared not-correctable [17]. The probability of an error passing undetected is less than 10^{-8} [2].

Mode S utilizes a shortened cyclic code to perform downlink error detection and correction. Cyclic codes are a type of parity encoding that are well suited to burst error channels [18]. The Mode S EDC code is also a systematic code, as the encoded transmission consists of the unmodified message followed by the parity field. The Mode S parity field consists of 24 parity bits. The Mode S EDC coding is described in detail in [17].

For an encrypted squitter application, encryption must take place prior to parity encoding, as shown in Figure 4.3-2 [11]. This is due to the error propagation associated with decryption. For example, should encryption take place after parity encoding, a single transmission error would propagate upon decryption, rendering the message useless. The single error would affect several bits due to confusion and diffusion associated with encryption, and the EDC code could not correct the error.

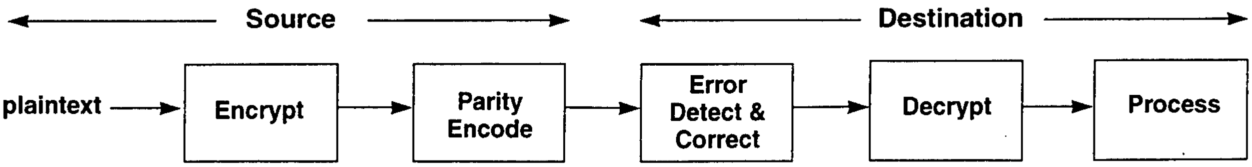


Figure 4.3-2. Required order for encryption and parity encoding. An error that occurs during transmission is correctable, allowing the receiver to correctly decrypt the received ciphertext.

When implemented as in Figure 4.3-2, it is imperative to note that EDC requires the parity bits to be appended in the PI field and broadcast in the clear (non-encrypted), as is consistent with systematic parity encoding. This reduces the payload available for encrypted ciphertext by 24 bits. However, the impact on squitter content should be negligible, as DF 17 also utilizes 24 bits solely for parity coding purposes. The requirement to broadcast 24 parity bits in the clear should not impact message content for encrypted squittering.

Encryption should have no impact upon the existing Mode S EDC scheme. The data passed to the Mode S parity encoder has no impact on EDC performance. The same processing steps will generate parity bits regardless of whether the data passed to the encoder is DF 17 AA + ME fields or ciphertext for an encrypted squitter.

Mode S parity encoding enables downlink error detection and correction in an error environment dominated by burst errors caused by synchronous garble. Error correction significantly reduces the probability of receiving incorrect messages and also reduces the need for re-transmission [17]. Moreover, without error correction, a single uncorrected bit-error would propagate during decryption, resulting in a useless squitter. For these reasons, Mode S EDC is necessary for encrypted squittering. The existing parity code should be used without modification, with emphasis that parity encoding must occur subsequent to encryption and prior to decryption, and that parity bits must be broadcast in the clear. Use of the existing code without modifications will provide robust EDC capability for encrypted squittering.

5. Encryption

This chapter will discuss feasibility of the application of encryption to Mode S ADS-B. As mentioned in Chapter 2, one method to determine the feasibility of a cryptosystem is to measure the system's ability to provide desired security services. Applying this approach to ADS-B, the following services will serve as metrics for the feasibility of encryption:

- Confidentiality. Encrypted squitters should be intelligible only to authorized users. Unauthorized parties attempting to intercept and decrypt squitters should be deterred by the complexity of such an attack.
- Identification. Each squitter should identify the source of the broadcast. Identification enhances surveillance by differentiating sources in the broadcast environment. Identification also simplifies authentication and integrity services.
- Authentication. The recipient of a squitter should be able to verify that the message source is an authorized or authentic user. Spoofing, or fabrication of spurious squitters meant to deceive authorized users, should be countered by authentication.
- Integrity. The cryptosystem should ensure that a received squitter has not been altered during or before transmission. Error detection and correction is a non-cryptographic source of integrity. However, additional mechanisms are necessary to ensure the integrity of squitters against active attacks.

To determine the feasibility of specific methods to provide the above metrics, the approach taken in this thesis is to analyze specific attacks. Figure 5-1 provides an overview of the problem. Security concerns for an ADS-B cryptosystem can be widely categorized into active and passive attacks [10]. During an active attack, adversaries interfere with the transmission of information. These attacks are a concern, because it may result in unsafe flight conditions due to the presentation of misleading information to pilots. Passive attacks are equivalent to eavesdropping, where an adversary listens and attempts to gather information from squitters. These attacks are difficult to detect and are dangerous because an adversary can gain insight into position, tactics, and strategy through surveillance information.

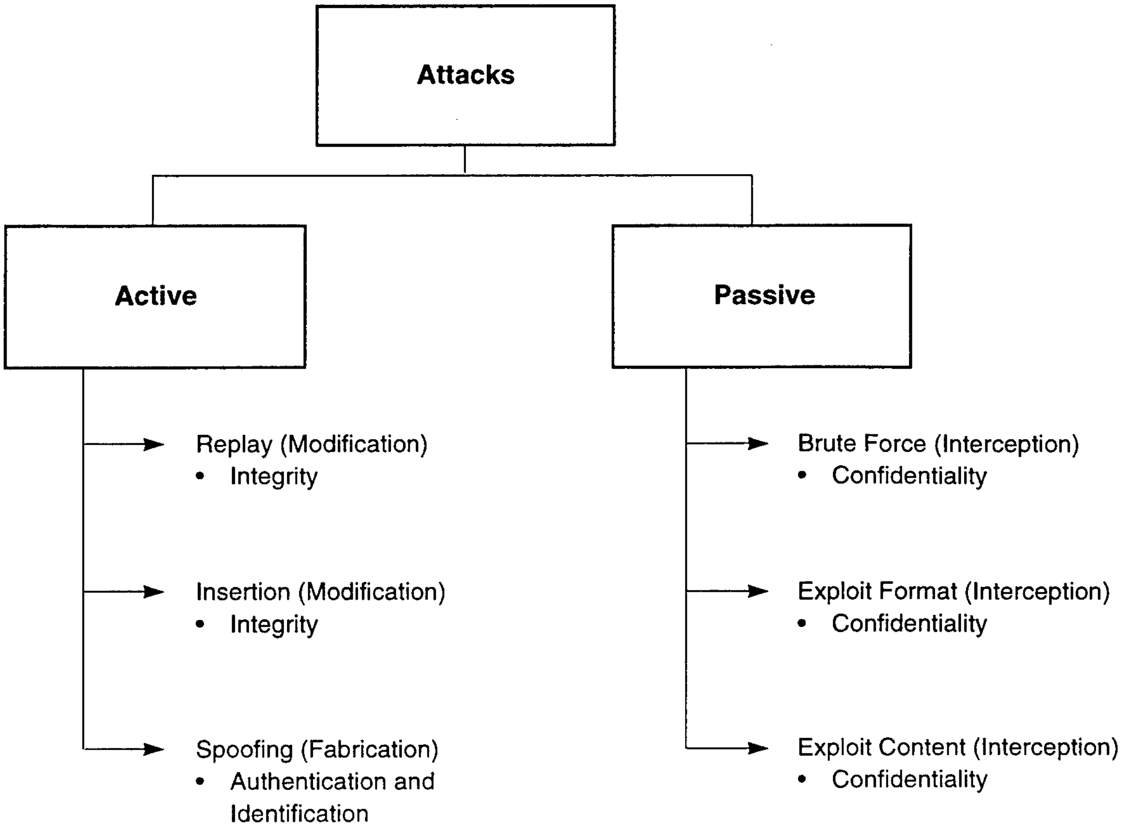


Figure 5-1. Summary of hypothetical attacks on a Mode S ADS-B cryptosystem. Attacks can be categorized as active or passive. Within each category, specific attacks create concerns for encrypted squitters, and these attacks will be discussed in detail in Chapter 5. In parenthesis next to each attack is the type of attack as introduced in Section 3.3. Under each attack is the security service that is required to prevent the attack.

The specific types of attacks listed in Figure 5-1 will be discussed in greater detail throughout Chapter 5, in no particular order. It is important to note that each attack can be countered by a specific security service. Equivalently, each attack exploits vulnerabilities in a specific service a system provides inadequately.

To simplify the discussion of feasibility, this chapter looks at specific methods to implement the cryptosystem. Section 5.1 provides a theoretical format for an encrypted squitter that will be used for the remainder of the thesis. In doing so, this section introduces some of the vulnerabilities due to formatting and squitter content that may be exploited by interception attacks. The format also includes the discrete Mode S aircraft address, which is consistent with DF 17 and provides identification without additional cryptographic complexity.

Sections 5.2 and 5.3 take a closer look at symmetric algorithm alternatives. Algorithm type and design are essential for ensuring confidentiality and deterring passive attacks. Section 5.2 covers stream algorithms by analyzing formatting vulnerabilities caused by repeated keystream segments and synchronization, which could result in weakness against interception attacks. Insertion attacks for stream algorithms are also briefly covered, as such attacks are unlikely but exacerbated by stream algorithms. Section 5.3 discusses characteristics of block algorithms. It also covers means of feedback that help make block algorithms resistant to passive attacks that attempt to exploit formatting patterns and repeated content of squitters.

Section 5.4 addresses authentication and integrity of encrypted squitters against active attacks. Spoofing (masquerade) and replay attacks are discussed. Methods to authenticate squitters and ensure their integrity are presented.

Section 5.5 is primarily concerned with implementation alternatives. This section discusses less-conventional methods to encrypt the squitter, given the short, irregular data length and format constraints.

Section 5.6 returns to the metric of confidentiality by discussing symmetric keylength concerns. Summaries of contemporary literature on keylength are presented, as well as implications on the impact of keylength on encryption for the lifetime of the ADS-B cryptosystem.

5.1 DF 19 Squitter Composition

ICAO has reserved Downlink Format 19 (DF 19) for use as an ADS-B military squitter [19]. One potential role of the DF 19 squitter is to broadcast encrypted aircraft data to other friendly, military aircraft without giving away location to adversaries, to provide situational awareness and help prevent mid-air collisions. Although DF 19 is reserved, its format is not yet defined. Format definition for DF 19 will likely be an evolutionary process based on users' needs, and the squitter should initially be flexible to accommodate a variety of military applications. This thesis does not attempt to define an ICAO standard or novel message content for DF 19. However, successful encryption must take into account the structure and format of the underlying plaintext, as patterns caused by formatting may simplify cryptanalysis. This section discusses format options and establishes a baseline format that will be used for the rest of the thesis.

5.1.1 DF 17 Civil Squitter Description

The extended squitter Downlink Format 17 (DF 17) is detailed in Section 4.2 and Appendix A, and it has five primary mission fields as described in Figure 5.1-1 [15]. This existing DF 17 squitter format used for civil ADS-B has evolved over many years. There are many civil aircraft types, resulting in a wide variety of navigation avionics capabilities. Moreover, Mode S equipage is gradual, resulting in disparity in transponder models and capabilities, as well as models produced by a variety of vendors. Data sources such as GPS units and altimeters also vary in precision. DF 17 remains flexible to accommodate such avionics variety, by creating subfields and squitter format type code specifiers to describe the information contained in each broadcast.

(5)	(3)	(24)	(56)	(24)
DF 17	CA	AA	ME	PI

DF 17 Extended Squitter

Figure 5.1-1. Bit breakdown and mission fields for the DF 17 ADS-B extended squitter. The number of bits for each field are shown in parentheses. DF – Downlink Format identifies the squitter type, here set equal to 17. CA – transponder capability describes the aircraft transponder capability. AA – address announced is each aircraft's unique Mode S address. ME – ADS message field contains aircraft state information for surveillance. PI – Parity overlaid with interrogator identity (II), where II is set to zero for squitter applications.

Airborne position reporting is one example of the flexibility built into the DF 17 extended squitter. There are 13 different format type code specifiers for airborne position, including 10 for altitude derived from barometric altimeters and 3 from GPS height above ellipsoid (HAE) as shown in Appendix A [14]. The 13 type codes account for differing horizontal position error based on own aircraft's navigation avionics capability.

Tactical military squittering should not require such a flexible approach. Civil users will disregard DF 19, and the DF 19 squitter format need not conform to the format used by civil avionics platforms. Although DF 19 should initially permit incremental development of several military applications, a greater level of hardware standardization should be assumed for military aircraft. For example, equipage with a common GPS receiver should allow standardized position and altitude reporting. Furthermore, U.S. military aircraft are scheduled to undergo a transponder upgrade to meet European Mode S equipage mandates, and to achieve Mode 5 IFF capability [19]. This procurement of new equipment could result in standardized equipage across many different types of tactical aircraft.

5.1.2 DF 19 Format Considerations

As DF 19 will be a military broadcast between platforms with greater hardware standardization, DF 19 composition should require fewer subfields and specifiers than DF 17. As a result, DF 19 users should take advantage of more bits for useful data. While it is noted here that more efficient bit use may be available for DF 19 applications, detailed research and design is needed to verify this. In addition, each military DF 19 user should research efficient bit use most desirable for their application.

Again, the emphasis for this thesis is DF 19 encryption rather than defining bit fields for DF 19. To further this objective and provide a plaintext basis, this thesis assumes DF 19 will utilize existing DF 17 formats for position and velocity squitters. Only minor modifications to the DF 17 format are suggested where necessary to enable encryption. This assumption would enable data compatibility with civil Mode S ADS-B services and also straightforward upgrade to DF 19 services.

Initial ICAO direction for DF 19 use suggests that the first 8 bits be broadcast in the clear [19]. Consistent with DF 17, the first 5 bits will identify the Downlink Format (DF), in this case equal to 19 (decimal). However, the remaining 3 bits constitute an Application Field (AF), replacing the DF 17 transponder capability (CA) field. Replacing the CA field with a DF 19 AF field is consistent with an incremental approach to allow gradual military use of DF 19 ADS-B. It also agrees with the notion of greater hardware standardization, as the transponder capability field should be unnecessary for military squitters.

This AF format allows up to 8 different DF 19 applications. AF 0 is already designated for a non-encrypted military squitter [19]. For the purposes of this thesis, assume that AF 1 is designated an encrypted squitter. Note that each application may operate independently, as the application field is broadcast in the clear, and non-authorized users of the encrypted squitter would disregard AF 1. Although practical considerations motivate a single encryption mechanism, each encrypted application should have freedom to chose independent keylength and key schedules to provide security for their application. For example, if AF 2 is reserved for experimental aircraft testing in restricted airspace, users of AF 1 would not be able to decrypt the experimental broadcasts. ICAO has not recommended any format for the remaining bits, and these bits should be designed subject to military requirements. Nevertheless, all DF 19 users must agree to the underlying breakdown of control, data / ciphertext payload, and parity for the remaining 104 bits.

Under the assumption that the encrypted squitter (DF 19, AF 1) will initially utilize existing DF 17 formats, the initial address announced (AA) and message (ME) fields are assumed to be identical to DF 17 formats. Furthermore, this also suggests the existing Mode S error detection and correction (EDC) scheme will be used for DF 19, which generates 24 parity bits. Therefore, the 80-bit combined AA and ME field should provide the basis for DF 19 plaintext data. Secure ADS-B requires that the aircraft address inside the AA field and the aircraft state parameters inside the ME field be confidential. As detailed in Section 4.3, the parity encoding must occur after encryption and parity bits must also be in the clear. This results in exactly 80 ciphertext bits per squitter. For purposes of this thesis, the default format for all DF 19 applications is an 80-bit message payload (MP) field, replacing the AA and ME fields, and a 24 bit parity field. For encrypted squitters, this permits an 80-bit ciphertext field (CF) replacing the MP field. This DF 19 field composition for an encrypted squitter is illustrated by Figure 5.1-2.

(5)	(3)	(80)	(24)
DF 19	AF 1	*Ciphertext Field*	PI

Figure 5.1-2. Suggested DF 19 bit composition for an encrypted squitter. The DF 17 CA field is replaced by a three-bit application field (AF), set equal to AF 1 to designate encryption. The AA + ME field is replaced by an 80-bit ciphertext field, indicated by italics. Within the ciphertext field, the encrypted aircraft Mode S address enables identification of the squitter's source.

With the plaintext message format identical to the DF 17 AA and ME fields, this DF 19 format enables identification. Upon decryption of the received squitter ciphertext, own aircraft will identify the source of the squitter through the plaintext AA field.

This DF 19 format approach assumes that the transponder capability field is not necessary for military applications, and that eight applications are sufficient for DF 19. However, ICAO is an international organization and DF 19 will also most likely be an international service. The CA field may be a necessary descriptor because avionics standardization among aircraft from different nations may be difficult. Likewise, if each nation desires an independent AF for cryptographic use, allocation for eight applications will be insufficient. Therefore, if these assumptions prove to be false, the AF must be relocated inside the squitter, and may require more than three bits.

An alternative approach is to overlay the AF and the PI field. In the existing extended squitter EDC scheme, the parity bits are exclusive-ored with the interrogator identity (II) field, which is zero

$$PI = parity \oplus II = parity\,.$$

A modification would allow overlay of the AF and II field before parity coding. Consider an exclusive-or of the II and AF at own aircraft, followed by parity coding

$$PI = Parity \oplus (II \oplus AF) = parity \oplus AF\,.$$

The AF could be larger than three bits, allowing more freedom than simply replacing the DF 17 CA field. This would allow retention of the transponder CA field, and could also permit independent use of the encrypted squitter for different users, to include unique algorithm and keylength choices. Upon receiving a DF 19 squitter, the EDC unit would generate a decoded address. If the decoded address equals the zero II field exclusive-ored with the AF, the receiving aircraft recognizes the squitter as a DF 19 squitter that matches the desired application. As shown in Figure 5.1-3, this approach requires a more complex error correction scheme. The error syndrome must recognize application formats, rather than a zero expected address. If an aircraft is utilizing more than one DF 19 application, multiple error correction steps would be required.

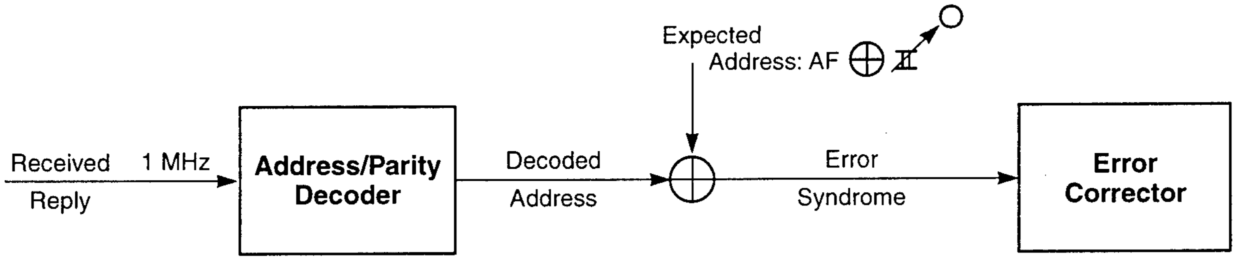


Figure 5.1-3. Processing of an overlaid application field (AF) and interrogator identity (II = zero) at a receiving aircraft transponder EDC unit. The incoming squitter is passed through an address/parity decoder which generates a decoded address. An exclusive-or compares the decoded address with the expected (II⊕AF) address. Error-free reception generates a zero error syndrome, which is passed to the error corrector.

5.1.3 Summary

DF 19 composition should initially mimic DF 17, to enable simple upgrade to DF 19, maintain compatibility with civil data formats, provide an identical EDC capability, and enable incremental integration for a variety of military applications. For this thesis, the AA and ME plaintext fields are assumed to be identical to those in civil DF 17 position and velocity squitters. This allows identification of the squitter source upon decryption. Standardized avionics could eventually allow more efficient bit-use within the DF 19 squitter, but additional research for specific military applications would be required to define modified data fields, and that is beyond the scope of this thesis.

To enable several military applications, a required modification to DF 17 is addition of an application field, which must be broadcast in the clear for encrypted squitters. The most straightforward means to accomplish this is to replace the DF 17 CA field with an AF field, allowing up to eight different DF 19 applications. This approach results in a standardized 80-bit message payload block for DF 19 applications, which must be encrypted to generate a ciphertext field for confidential communication. This approach does not require modification to the Mode S error detection and correction scheme. The remainder of this chapter will discuss encryption of the 80-bit DF 19 ciphertext field.

5.2 Case Against Stream Algorithms

Symmetric encryption algorithms fall into either stream or block algorithm types. The types are loosely differentiated from one another by the number of input bits the algorithm processes at a given time. Stream algorithms can be further differentiated as synchronous or self-synchronizing algorithms. This section analyzes the use of stream algorithms for DF 19, by focusing on specific attacks against the cryptosystem.

A stream cipher is a symmetric algorithm that operates on streams of plaintext, usually one bit at a time. Chapter 3 of this thesis described how symmetric algorithms require the sender and receiver to possess the same symmetric key. The security of the stream cipher depends on the strength of the keystream generated with the key. Although the keystream is deterministic, an adversary will have a difficult time compromising the keystream if it is very-nearly random [11].

5.2.1 Self-Synchronizing Stream Algorithms

Self-synchronizing stream algorithms synchronize keystreams using a predetermined header and a fixed number of previous ciphertext bits fed back into the algorithm [11]. The destination receiver decrypts the required number of ciphertext bits and passes them back to the algorithm which synchronizes the internal state of the algorithm. This header is decrypted incorrectly and discarded. However, this results in a synchronized keystream, and ciphertext bits subsequent to the fixed-length header are decrypted correctly.

The use of a header as well as extensive ciphertext feedback are not desirable for squittering. The air-air link is not continuous and each DF 19 squitter must be independent of previous squitters to allow for squitters that are lost due to interference or range. The short 80-bit block would not allow the necessary iteration inherent to self-synchronizing algorithms without excessive complexity and initialization of each squitter with the same header. Failure to utilize several rounds of iteration would weaken the encryption, as patterns would arise making the system susceptible to passive attacks. The header would also result in otherwise useless overhead bits. Self-synchronizing stream algorithms are not considered further in this thesis.

5.2.2 Synchronous Stream Algorithms

As shown in Figure 5.2-1, both the source and the destination side generate identical, synchronized keystreams for synchronous stream algorithms [11]. The symmetric mission key is a parameter for the output function, which generates the keystream. The internal state may require an initialization vector, which is an initial state the algorithm returns to at the start of keystream generation. The next state function is typically a simple function such as a counter that updates the internal state. The resultant ciphertext is a simple exclusive-or of the plaintext and the keystream, which is reversed for decryption at the destination.

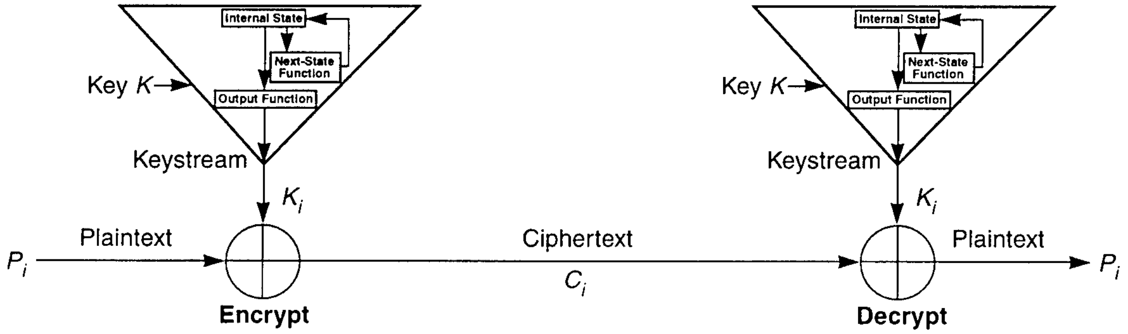


Figure 5.2-1. Synchronous stream algorithm operation.

Because the keystream generator is finite-state and therefore periodic, the keystream period must be much greater than the length of the plaintext to be encrypted before re-keying [11]. Otherwise, different plaintext segments will be encrypted identically, enabling the cryptanalyst to essentially remove the keystream from a passive attack:

Original plaintext #1:	$p_1\ p_2\ p_3\ p_4$	Original plaintext #2:	$p_5\ p_6\ p_7\ p_8$
⊕ Original keystream:	$k_1\ k_2\ k_3\ k_4$	⊕ Repeated Keystream:	$k_1\ k_2\ k_3\ k_4$
Ciphertext #1:	$c_1\ c_2\ c_3\ c_4$	Ciphertext #2:	$c_5\ c_6\ c_7\ c_8$

With both ciphertext segments generated from identical keystream segments, the cryptanalyst simply combines the ciphertext which reduces the attack to a combination of plaintext segments.

$$\mathrm{C}_1 \oplus C_2 = (P_1 \oplus K_1) \oplus (P_2 \oplus K_1) = (p_1 p_2 p_3 p_4) \oplus (p_5 p_6 p_7 p_8) \qquad (5.2\text{-}1)$$

This simplifies analysis considerably, as plaintext has defined formatting and patterns, whereas keystreams are nearly random.

Another potential vulnerability of stream algorithms is malleability, which can be exploited by modification attacks that reverse bits [20]. Modification is an active attack, where the adversary attempts to tamper with the ciphertext during transmission. Such an attack does not appear feasible for ADS-B. Although an adversary could attempt to insert energy bursts into the squitter environment, the Mode S error detection and correction scheme would correct the burst errors or discard the message before deciphering. Nevertheless, synchronous stream algorithms do not diffuse patterns in the plaintext or rearrange bit order, which makes the ciphertext more malleable. This characteristic of stream algorithms should be considered undesirable for ADS-B, as it weakens the integrity of squitters.

5.2.3 Time Synchronization Requirement

The primary weakness of synchronous stream algorithms for squittering applications is the time synchronization requirement. As illustrated in Section 4.1, encrypted squittering will require interaction between the GPS receiver, the transponder, and the ADS-B processor.

Figure 5.2-2 details the timing during the steps required to generate encrypted DF 19 position squitters under the assumption of encryption with a synchronous stream cipher [12]. A similar process was described for the DF 17 squitter in Section 4.2.

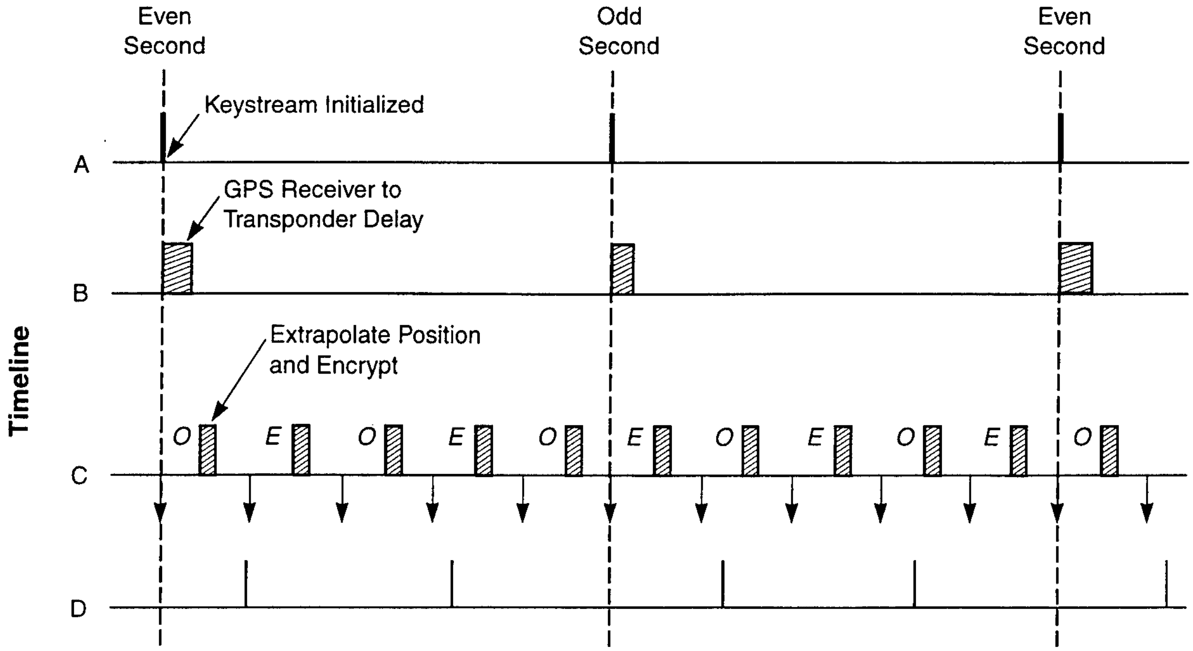


Figure 5.2-2. DF 19 squitter position update timelines, with transponder clock synchronized with GPS time. Timeline A shows GPS position, velocity, and time (PVT) updates, as well as synchronized re-initialization of the stream algorithm keystream. Timeline B shows completion of the position and velocity transfer to the transponder. Timeline C indicates the GPS-synchronized transponder clock ticks, after which the position is extrapolated, encrypted, and downloaded to transponder registers. The arrows indicate the extrapolated GPS position time of validity. Timeline D is the position squitter trigger.

The assumption in this study is that the transponder timing subsystem and therefore the ADS-B processor is synchronized with the GPS clock, which provides time mark pulses on each GPS second and is specified to be accurate within 120 ns [13]. This would also allow synchronization of the stream cipher keystream for all aircraft broadcasting encrypted squitters. The first step of the interaction required to produce DF 19 squitters is GPS synchronization, as shown by Timeline A in Figure 5.2-2. The GPS receiver generates position, velocity, and time (PVT) updates and time mark pulses every GPS second. Assume that the symmetric key is updated periodically as indicated on Timeline A, which also reinitializes the initialization vector (IV) and begins production of a new keystream. The symmetric key and associated keystream endure for several GPS seconds, and then are updated based on a predetermined key schedule.

Timeline B indicates complete download of the PVT update to the transponder, which can take from 4 to 200 ms [13]. The left edge of the bars above Timeline C indicate the transponder's GPS-synchronized 200 ms clock ticks, which alternate even (E) and odd (O) in accordance with the compact position reporting (CPR) format described in Section 4.2.

On each transponder clock tick, GPS position is extrapolated using GPS velocity to a position that is valid at time 100 ms after the transponder clock tick. For DF 19, after extrapolating the position, the ADS-B processor would concatenate the 56-bit ME field to the Mode S address, and encrypt the AA+ME field with the keystream, generating the 80 bit ciphertext field (CF). Once accomplished, the ciphertext would be uploaded to transponder registers. The delay associated with extrapolation, encryption, and upload is illustrated by the bars on Timeline C. The position ciphertext would be valid approximately 100 ms prior to the next transponder time tick, as shown by the down arrows below Timeline C. Under this scheme, the extrapolation, encryption, and upload steps must take less than 100 ms. Also note that each extrapolated position update will be encrypted, but not all updates will be broadcast.

Timeline D is the transponder position squitter trigger. The trigger occurs on average twice a second, with a randomized time interval of 400 to 600 ms between triggers. If the squitter triggers were not randomized, all aircraft would broadcast at nearly the same instant under the assumption that the broadcasts are synchronized with GPS time. This would create interference between aircraft that could not be overcome. Once triggered, the transponder then appends the eight format bits and zero II field, parity encodes the squitter, appends the 24 parity bits, and broadcasts the message. Mode S-equipped aircraft then receive the broadcast message, error detect and correct, recognize DF 19, and process the message if authorized. Herein lies the difficulty with using stream ciphers. The receiving aircraft have to be able to synchronize the received squitter with the correct keystream bits that generated the ciphertext. Without knowledge of the point in the keystream used for encryption, the receiving aircraft cannot decrypt the message.

Own aircraft could attempt to synchronize the keystream with other aircraft by segmenting the keystream and saving 80-bit segments on each 200 ms transponder clock tick. Own aircraft would then synchronize the first extrapolated payload bit to be encrypted with the first bit of the captured keystream segment, and proceed to encrypt all 80 bits. Under this method, the captured keystream would be valid until the next transponder clock tick. This method is not feasible, due to the broadcast nature of the problem and the repeated-keystream attack described by equation (5.2-1) above. All squitters broadcast during a given interval would use the same keystream, greatly simplifying cryptanalysis. Essentially, this approach equates to broadcasting data exclusive-ored with a fixed stream that updates every 200 ms. The adversary could compare all ciphertext intercepted during a 200 ms interval because the stream cipher would not diffuse plaintext bit dependencies. The adversary would not have to compromise the keystream or underlying mission key in order to obtain information. Moreover, the adversary's problem is simplified even more by recognizing patterns in the ciphertext caused by the structure of the payload sub-fields, which could be exploited to reveal segments of the keystream. Due to this keystream synchronization problem, synchronous stream algorithms do not appear to guarantee confidentiality.

The nature of the Mode S link and data format places synchronization constraints on the use of stream algorithms. Furthermore, stream algorithms do not permute plaintext bits, which leaves the ciphertext malleable and more susceptible to patterns in the absence of a strong keystream. The use of synchronous stream algorithms for DF 19 squitters could create weaknesses in confidentiality and integrity for the encryption system.

5.3 Block Algorithms and Modes

The alternatives to symmetric stream algorithms are block algorithms. Block algorithms offer advantages over synchronous stream algorithms because they help diffuse bit dependencies, which makes them less susceptible to patterns. Symmetric block algorithms are usually implemented in a cryptographic mode which uses simple feedback and other operations to help ensure the security of the system. This section analyzes the use of block algorithms and modes for ADS-B by focusing on specific attacks on the DF 19 squitter.

Symmetric block algorithms usually operate on 64-bits of plaintext, or multiples thereof, as opposed to bit-wise or byte-wise operations with stream ciphers [11]. A significant advantage of block algorithms over stream algorithms is diffusion, which spreads the influence of individual plaintext bits over as much of the ciphertext as possible [11]. For DF 19, the lack of diffusion in synchronous stream algorithms is a disadvantage, as each plaintext bit would contribute to only one ciphertext bit through a simple operation with the keystream. The block size for the algorithm must be less than or equal to 80 bits. However, 80 bit algorithms are generally non-existent or inaccessible for public use.[2] Symmetric 64-bit block algorithms abound and are an alternative to 80-bit algorithms.

There are a variety of very strong 64-bit symmetric block algorithms publicly available. The ability of these algorithms to provide confidentiality is often proven by withstanding scrutiny from the cryptanalyst and hacker communities. Once again, it is emphasized that DF 19 encryption should utilize an algorithm that has stood up against such scrutiny. A characteristic of many strong block algorithms is multiple rounds of encryption, as well as S-box design that is resistant to cyptanalysis. During each round, S-boxes implement substitutions, or mappings of m-bit inputs to n-bit outputs. Figure 5.3-1 shows the S-box array for DES, where each S-box receives a 6-bit input, which is replaced by a 4-bit output value [11]. The algorithm chosen should also take into consideration the avalanche effect for S-box design. Strict avalanche criteria (SAC) for S-boxes ensures that if only one input bit is changed, exactly one half of the corresponding output-bits change [11]. This helps ensure confidentiality by removing ciphertext patterns despite patterns in the plaintext. A proven algorithm with S-boxes that satisfy SAC is desirable.

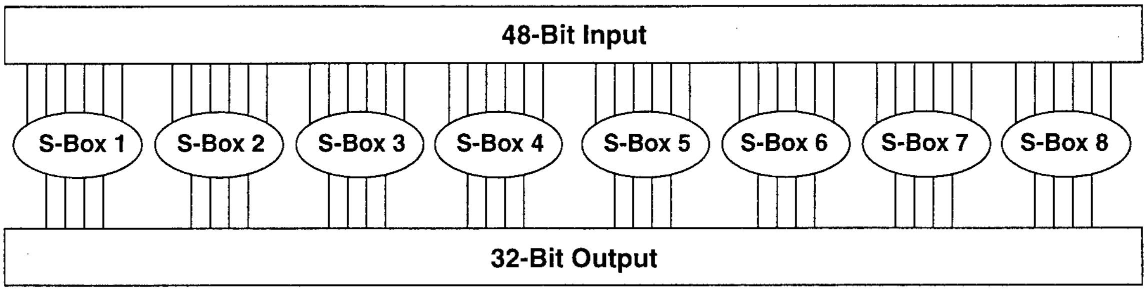


Figure 5.3-1. S-box substitution array for DES. DES uses 8 S-boxes, and substitution occurs through the S-boxes in each of the 16 rounds.

5.3.1 Cryptographic Modes

To strengthen the security of the underlying algorithm against certain types of attacks, implementation of symmetric block algorithms should incorporate a cryptographic mode. Modes combine feedback and simple operations with the basic algorithm [11]. Implementation of block algorithm modes essentially strengthens the encryption scheme by concealing patterns in the plaintext that may result in ciphertext patterns. However, because the operations and feedback

[2] One exception is REDOC II, which is a patented 80-bit block algorithm.

are simple, the security of the system against brute-force attacks remains a function of the underlying algorithm and key, and not the mode [11].

Interception is a passive-listener attack where an adversary intercepts ciphertext attempting to exploit patterns, and it is the primary type of attack that modes defend against. More specifically, the mode of operation should strengthen DF 19 against inherent weaknesses caused by the structure of the AA+ME plaintext and also the repetitive nature of squittering several times per second. Both of these weaknesses could cause patterns in the ciphertext that an adversary could exploit if not addressed by the DF 19 cryptographic mode.

In addition to diffusing DF 19 plaintext patterns to strengthen confidentiality, the mode must meet other objectives stated earlier. Similar to requirements for the underlying algorithm, after implementing a mode the ciphertext length must equal the 80-bit plaintext length with no overhead bits, and each squitter should stand alone. The mode must not interfere with Mode S error detection and correction. Additional encryption steps as well as any feedback included in the mode should nonetheless enable efficient encryption and decryption, to enable the required squitter throughput. That is, a simple and efficient mode is advantageous.

The DF 19 squitter encryption mode must also consider timing and synchronization issues. Depending on system requirements, the duration of a single symmetric key may be on the order of tens of seconds to minutes. Under this configuration, a single key will encrypt several squitters, as squitters are broadcast up to 6.2 times per second. If the transponder timing subsystem and ADS-B processor are synchronized to a GPS clock, aircraft can synchronize key updates based on a predefined key schedule. The mode should enable own aircraft decryption to unambiguously match received DF 19 squitters from other aircraft to the correct key that encrypted the broadcast, as well as other parameters such as initialization vectors or counters.

The application of block algorithm modes usually follows one of four conventional methods [11]:

- Electronic Code Book (ECB) mode is a straightforward implementation of the block algorithm, without feedback. ECB is most vulnerable to patterns and redundancies, because the adversary can obtain information by accumulating a code book or look-up table of ciphertext fragments and recognize fragments that repeat, without obtaining any knowledge of the key.

- Cipher Block Chaining (CBC) incorporates feedback by including previous ciphertext results with the encryption of the current plaintext block, so that each ciphertext block depends both on the current plaintext block and all previous blocks.

  $$\text{Encryption: } C_i = E_K(P_i \oplus C_{i-1})$$

  $$\text{Decryption: } P_i = C_{i-1} \oplus D_K(C_i)$$

  A disadvantage of CBC is that similar messages encrypt the same way until the first difference in the plaintext. To overcome this, an initialization vector (IV) or initial state serves as the initial plaintext block at the start of the encryption process to ensure that each ciphertext is unique. The IV should be unique for each message, but it need not be random or secret. A counter or timestamp is sufficient.
- Cipher Feedback (CFB) is a mode that implements a block cipher as a self-synchronizing stream cipher. Essentially, it uses the block algorithm to generate the keystream, and relies on ciphertext fed back into the algorithm to synchronize the internal state. CFB also requires an IV to begin keystream generation, but in contrast to CBC, the IV must be unique for every message.
- Output Feedback (OFB) mode implements the underlying block cipher as a synchronous stream cipher. It is a means of internal feedback, so that the keystream does not depend on previous plaintext or ciphertext. As such, OFB is very fast, and the keystream can be generated off line. Counter mode is an OFB variant, which uses a counter to generate the internal state rather than keystream feedback. The IV should also be unique for every message.

As described in Section 5.2, stream implementations such as CFB and OFB would create weaknesses for the DF 19 cryptosystem. In general, the short 80-bit block makes self-synchronization with CFB a poor choice for DF 19. For CFB, all aircraft would begin with the same IV, so that similar messages would encrypt the same way until the first difference in the plaintext. Each squitter would require re-initialization, which increases memory requirements and synchronization complexity. Such an application would also be considered poor technique for CFB, which is meant to provide an extended keystream requiring several encryption steps with ciphertext feedback on each step. However, this would be difficult to accomplish for DF 19, as each squitter must stand alone, because some squitters may not be correctly received.

With OFB, synchronization difficulties from stream implementations of the underlying block algorithm could result in repeated ciphertext segments and a weakened crypto-system. OFB operates essentially as a synchronous stream cipher. Due to the nature of the air-air broadcast the cryptanalysts could remove the keystream from an attack, as all aircraft would use the same keystream segment to generate ciphertext. OFB is also malleable, and plaintext is not permuted with respect to the keystream.

5.3.2 Requirements for ECB and CBC Modes

ECB and CBC are the remaining mode choices for DF 19. The primary objective for mode implementation is increased confidentiality through the prevention of ciphertext patterns caused by the structure of DF 19 plaintext and the nature of squittering several times per second. Two scenarios create plaintext patterns that the algorithm mode must address. The first creates redundancy in squitters between different aircraft, and an adversary could exploit this with a "stereotyped-beginnings" attack. The second scenario creates patterns in consecutive squitters from own aircraft, and an adversary can exploit this vulnerability through a "repeated-plaintext" attack.

Stereotyped beginnings occur when plaintext has a well-defined structure such as headers that create a pattern in every message [11]. In a stereotyped-beginnings attack an adversary would exploit the DF 19 structure, looking for patterns in ciphertext resulting from the message field (ME) plaintext structure. For example, a DF 19 position squitter could utilize a well-defined ME field, such as the DF 17 position field shown in Figure 5.3-2 [14]. The position ME field for aircraft in formation may differ only slightly, and all aircraft will utilize the same symmetric key. If several aircraft were at the same altitude in formation, bits 1 through 22 could be identical, creating a defined header for the position squitter. Similar patterns would arise for velocity and other squitter types.

1	
2	FORMAT TYPE CODE
3	
4	
5	
6	SURVEILLANCE STATUS
7	
8	SINGLE ANTENNA FLAG (SAF)
9	
10	
11	
12	ALTITUDE (barometric or GPS HAE, as specified by
13	the format type code)
14	
15	
16	
17	
18	
19	
20	
21	TIME SYNCHRONIZATION (T)
22	CPR FORMAT (F)
23	MSB
24	
25	
26	
27	
28	
29	ENCODED LATITUDE
30	(CPR airborne format)
31	
32	
33	
34	
35	
36	
37	
38	
39	LSB
40	MSB
41	
42	
43	
44	
45	
46	ENCODED LONGITUDE
47	(CPR airborne format)
48	
49	
50	
51	
52	
53	
54	
55	
56	LSB

Figure 5.3-2. DF 19 ME plaintext field for a fixed airborne position type squitter.

To address the stereotyped beginning, DF 19 should rely on the strength of the underlying algorithm and key. Although the mode can also help remove patterns, avalanche effect for the DF 19 algorithm should ensure a secure message if only one bit changes in the plaintext. Therefore, to prevent any correlation between aircraft squittering in close proximity, the first plaintext block into the DF 19 algorithm should include the aircraft's AA field. This 24-bit Mode S address is unique for each aircraft. As a result, the relationship of the first ciphertext block from one aircraft to another is guaranteed to be blurred, despite the occurrence of identical ME bits. Feedback to generate the remaining 16 ciphertext bits further ensures the confidentiality of the entire ciphertext block.

The second pattern DF 19 must guard against arises during repeated squittering by a single aircraft. Consecutive same-type squitters may generate ciphertext from plaintext that does not change between squitters, creating a repeated plaintext problem. A disadvantage of block algorithms is that repeated plaintext encrypted with the same key will result in identical ciphertext. An adversary could recognize and take advantage of identical ciphertext segments. This attack is dangerous, because several aircraft parameter bits inside the ME plaintext may remain constant as demonstrated by stereotyped beginnings. In addition, the AA plaintext is constant for own aircraft. Consider a helicopter hovering so that its GPS position remains constant, or a fixed-wing aircraft on a constant velocity vector so that its velocity squitter changes very little from broadcast to broadcast. If the underlying algorithm and cryptographic mode do not diffuse the similarities in repeated plaintext, the adversary could recognize a flight profile from redundancies in squitters without obtaining key information. Again, the solution for repeated plaintext should take into account the diffusion and avalanche effect characteristic of block algorithms. In this example, repeated squittering for own aircraft does not guarantee at least one bit change over a 64 bit segment. The mode must guarantee variable plaintext from squitter to squitter.

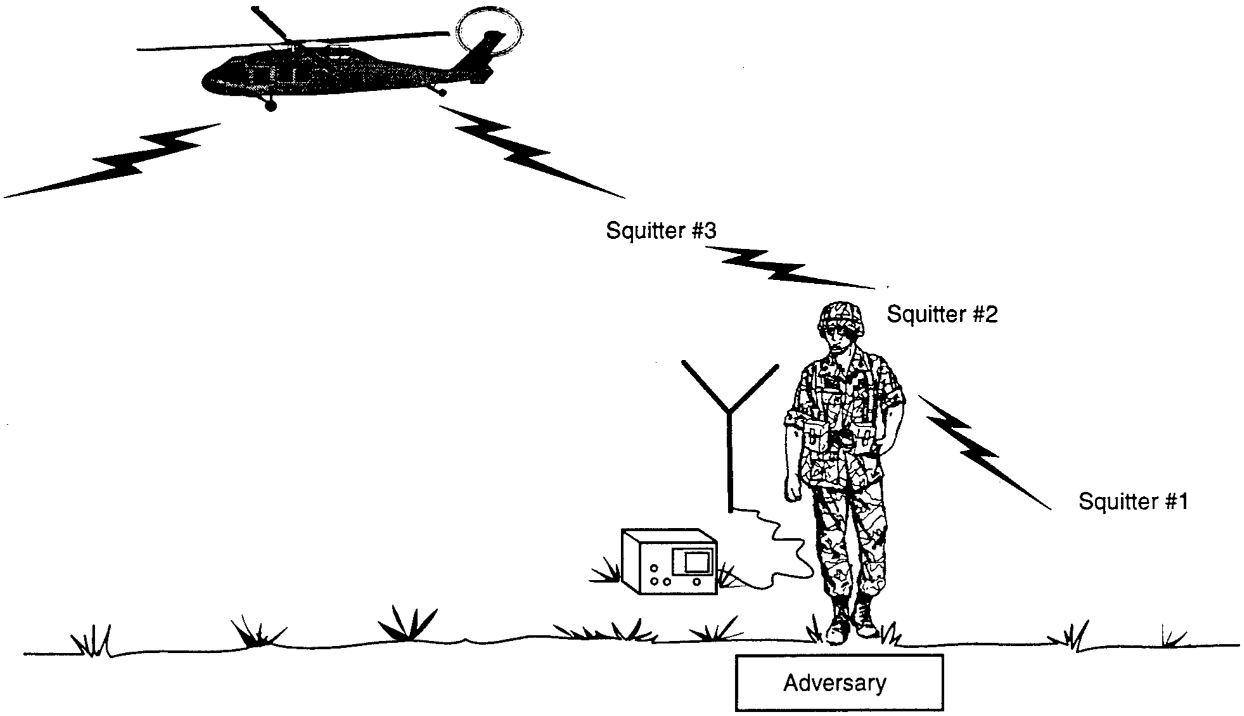


Figure 5.3-3. Repeated plaintext scenario: A hovering helicopter will squitter position with plaintext that changes only marginally between broadcasts, creating repeated plaintext and a potential advantage for the adversary cryptanalyst. In this scenario, the adversary can gain insight into the aircraft's flight profile without decrypting the squitter.

To elaborate upon the vulnerability caused by repeated plaintext, consider encryption of the 80-bit DF 19 plaintext with a 64-bit symmetric block algorithm. Methods to encrypt an 80-bit block with a 64-bit cipher will be discussed in detail in Section 5.5, but it is sufficient to note that the plaintext block will have to be segmented, and at least two encryption steps must be performed. Consider a scenario where a single aircraft is squittering and consecutive squitter plaintexts vary only in the final 16 bits of the ME field. Given a fixed mission key, if the ciphertext is segmented so that the first 64 bits correspond to the first 64 plaintext bits,

$$c_0c_1c_2...c_{63} = E_k[p_0p_1p_2...p_{63}]$$

and the remaining ciphertext bits depend on some combination of the remaining 16 plaintext bits and the preceding 64 plaintext or ciphertext bits, then the first 64 bits of the broadcast ciphertext

$$C_m = [c_0c_1c_2...c_{79}]$$

will not vary from squitter to squitter. As a result, the weakened encryption scheme simplifies analysis because of the pattern in consecutive ciphertexts

$$C_m\big|_{bits=0\ldots63} = C_{m+1}\big|_{bits=0\ldots63}.$$

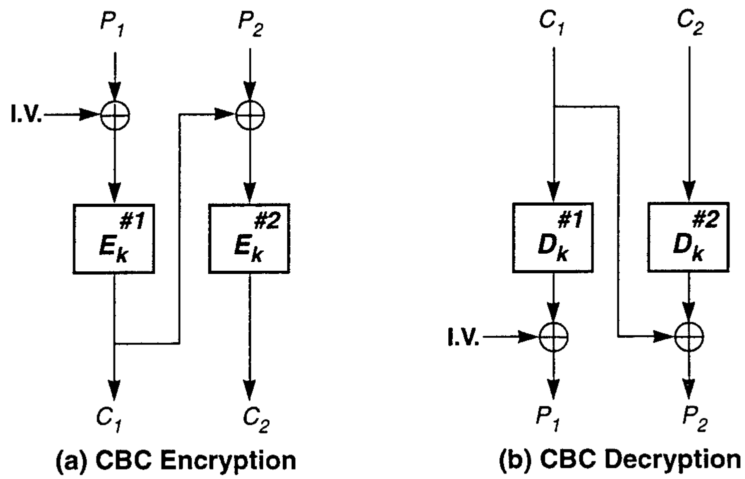
The system's confidentiality is therefore compromised without requiring the adversary to perform any cryptanalysis on the squitter.

The most obvious solution to repeated plaintext is to include an initialization vector that changes for each same-type squitter. For example, position and velocity squitters are broadcast on average twice a second, with a uniform distribution of 400 to 600 ms [14]. The IV could update on every other 200 ms transponder clock tick, where the transponder is synchronized to GPS time. A simple counter would be one example. Based on the defined key schedule, symmetric keys would update periodically. On each key update, the counter would reset to one and then increment every 400 ms until the next key update. Under this scheme, encryption for consecutive squitters of the same ME format type code would have a different IV, which would remove the repeated plaintext vulnerability. Although dissimilar DF 19 squitters could utilize the same IV and key during a 400 ms interval, the differing format type code for the squitters would result in variable plaintext.

5.3.3 Trade-off Between ECB and CBC Modes

ECB is the simplest cryptographic mode. A strong algorithm in ECB mode with a secure key provides variability and secure ciphertext if just one dissimilar plaintext bit is present. However for DF 19, ECB is susceptible to repeated plaintext attacks. Even with a proven 64-bit algorithm with S-boxes that satisfy SAC, ECB without feedback cannot conceal repeated 64-bit plaintext segments during a given key interval. If one bit does not change in the first encryption block of adjacent squitters, the first 64 ciphertext bits will be identical.

CBC mode includes ciphertext feedback. If used in conjunction with an IV that varies at least as frequently as same-type squitter broadcasts, CBC protects against repeated plaintext. Like ECB, CBC also benefits from diffusion and confusion of the underlying algorithm which removes any pattern existing in the ME field of DF 19. As shown in Figure 5.3-4, CBC is very efficient [11]. It requires only two encryption steps, an IV, and exclusive or operations for ciphertext feedback.



(a) CBC Encryption

(b) CBC Decryption

Figure 5.3-4. Cipher block chaining mode for DF 19. The process begins at the source with an initialization vector, which is equivalent to an initial plaintext block P_0, and it can be deterministic such as a counter or shift register. To prevent repeated-plaintext similarities for same-type squitters generated by own aircraft, the IV should vary at least every 400 ms. The IV exclusive-ored with the first plaintext block P_1 is encrypted which generates the first ciphertext block C_1. C_1 is fed back through an exclusive-or with the next plaintext block P_2 and encrypted resulting in the second ciphertext block C_2. The ciphertext blocks are combined and broadcast to the destination where the process is reversed.

5.3.4 Summary

The algorithm mode for DF 19 must ensure the confidentiality of the underlying algorithm by concealing plaintext patterns. Moreover, the mode choice should be simple and efficient in memory. Synchronous stream algorithms and modes such as OFB do not conceal ME patterns. The short data length for the extended squitter makes self-synchronizing modes such as CFB a poor choice for DF 19, due to additional complexity and synchronization requirements. ECB also fails to conceal repeated plaintext patterns as it is a very basic mode, without any kind of feedback. The security against plaintext patterns makes CBC with an initialization vector an effective mode choice for encrypted DF 19 squitters.

5.4 Authentication and Integrity

Authentication is a necessary service for the DF 19 encryption system, as only authorized users with correct symmetric keys should be able to send and receive the encrypted squitters. Authentication should also ensure with very high probability that a received squitter is from an authorized aircraft. Likewise, integrity service should prevent modification attacks, where an adversary may generate fraudulent messages. This section will discuss viable methods to provide message authentication and integrity for DF 19 by focusing on specific attacks known as masquerade and playback attacks.

5.4.1 Authentication and Integrity Through Encryption

Message authentication verifies that a message comes from the alleged source and also ensures the integrity of the message content [10]. Conventional encryption itself provides a measure of authentication in addition to confidentiality, and the ciphertext serves as the authenticator [10]. Consider DF 19 encryption, where only authorized aircraft have access to the symmetric keys. Decryption of a DF 19 squitter with the correct key will result in recovered plaintext. The receiving aircraft is assured that the message is authentic because only authorized users have key access, and the valid ciphertext authenticates the squitter's source and aircraft identity by decrypting the squitter into useful information. As shown in Figure 5.4-1, an adversary cannot generate repeatable ciphertext that decrypts to sensible plaintext without knowledge of the key. Therefore, the receiving aircraft must be able to differentiate legitimate plaintext from garbage in order to use ciphertext for authentication.

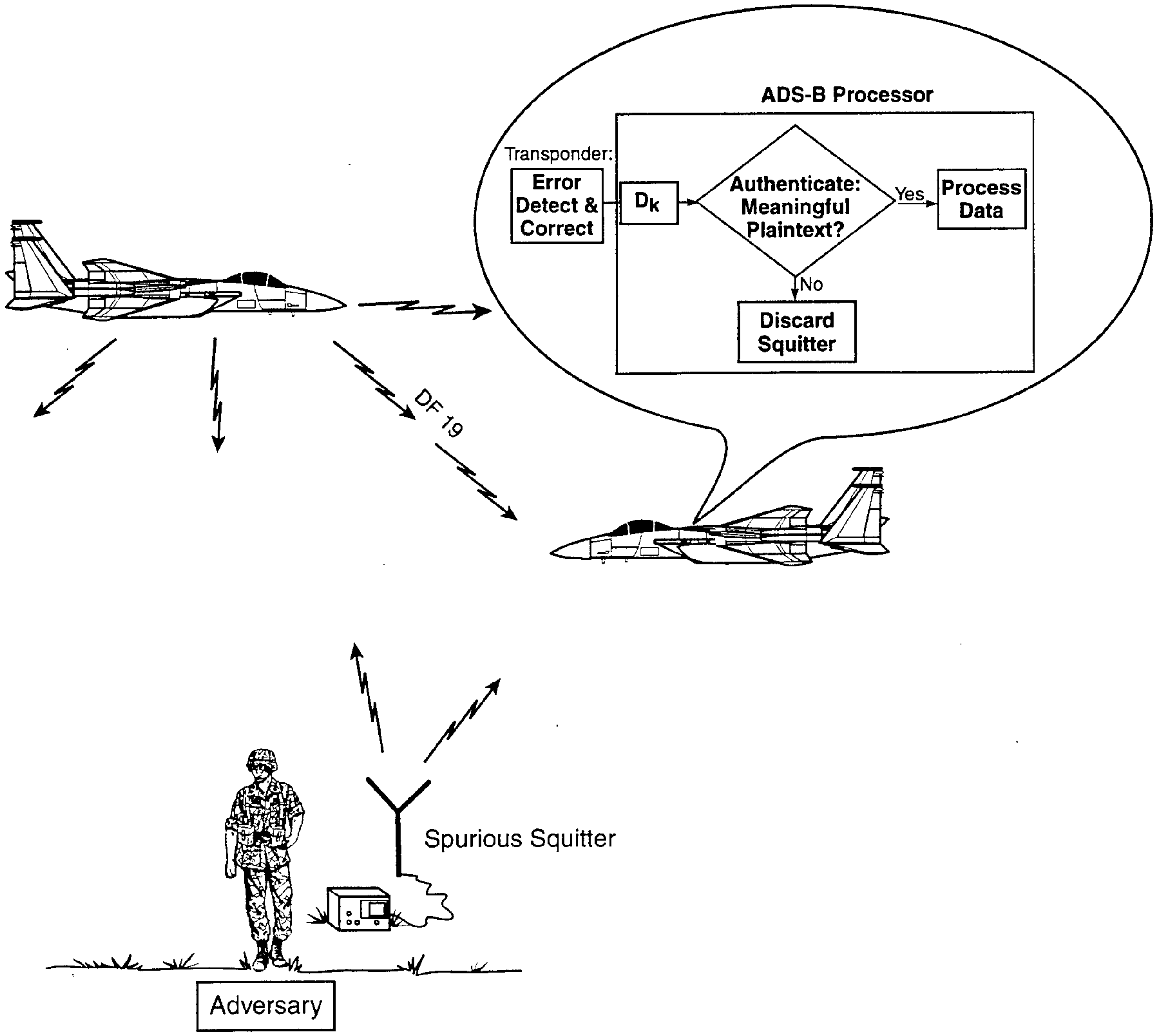


Figure 5.4-1. Ciphertext used as an authenticator. An adversary may generate garbage ciphertext. The aircraft receives and decrypts all squitters assuming they are legitimate. However, garbage ciphertext decrypts to meaningless information and is not authenticated. Conversely, a legitimate DF 19 broadcast is recognized and forwarded for additional processing.

5.4.1.1 Authentication and Masquerade Attacks

The DF 19 environment may be subject to spoofing or fabrication by an adversary. One type of fabrication attack involves insertion of fraudulent messages into the environment by a source masquerading as an authorized user [10]. An adversary without knowledge of the active symmetric key could attempt a masquerade attack by first generating spurious ciphertext. The adversary would then format the fraudulent DF 19 message, perform error coding, and broadcast the message. The receiving aircraft would decrypt the spurious ciphertext with the correct symmetric key. Of course, this ciphertext would likely decrypt into meaningless bits. Nevertheless, the adversary could disrupt the ADS-B processor by broadcasting several fraudulent squitters, as each received squitter will require EDC, decryption, and some method to determine if the received squitter is authentic. The authentication method chosen should simplify this third step, so that the disruption caused by a masquerade attack is minimized.

The ADS-B processor decrypts ciphertext, generating the plaintext message M, which must be authenticated. Note that if M can be any bit pattern, then spurious ciphertext would always generate valid plaintext, and the user would have to accept any message as authentic ciphertext [10]. However, for DF 19, the plaintext M follows a well-defined format based on the ME field. Furthermore, consecutive DF 19 messages received from the same aircraft will have identical AA fields and the surveillance data in the ME will be related. The DF 19 message should be authenticated based upon processing the AA and ME fields. Upon reception of a DF 19 squitter, the following steps illustrate one method to accomplish authentication:

- Error detect and correct the received squitter.
- Recognize Downlink Format = 19 and Application Field = 1.
- Decrypt the 80-bit ciphertext.
- Authenticate the address: Does the AA field match the address of a previously-received encrypted squitter? If yes, then the message is authentic.
- Authenticate the message content: Does the ME field make sense? Is the Format Type Code defined? For a position type squitter, is the aircraft within realistic range, etc.?
- If message content is valid, save the message for additional address authentication: Do not discard the squitter, but start tracking the aircraft and wait for reception of another encrypted squitter with the same AA field to complete authentication.

This process could be simplified by removing the message content authentication step. Encrypted squitters would only be authenticated after receiving at least two squitters with the same AA field.

5.4.1.2 Integrity and Replay Attacks

For civilian ADS-B, the integrity of DF 17 squitters is determined by error detection and correction. A squitter that is correctable, or does not require correction, is considered legitimate.

Assuming a benign environment with no spoofing, erroneous squitters occur only in the event of incorrect error correction, which occurs with a very low probability. Thus, for civil ADS-B applications, an aircraft transponder can determine with fairly high confidence that error-corrected DF 17 squitters are legitimate.

However, as illustrated with masquerade attacks, the DF 19 environment may be less benign and susceptible to active attacks. An additional fabrication attack is timing modification, also known as a replay attack, where an adversary records valid DF 19 squitters and replays them after some delay [10]. For example, an adversary could record DF 19 squitters during a training exercise. In an attempt to disrupt subsequent training or cause an accident, the adversary could then re-broadcast the squitters at a later time when other aircraft occupy the same airspace. Once again, conventional encryption can provide integrity against this attack if implemented properly. For example, the key validity interval (KVI) or key duration and the cipher block chaining (CBC) initialization vector (IV) would provide message integrity and guarantee that replayed ciphertext decrypts to garbage.

As illustrated in Section 5.3, one method to accomplish DF 19 encryption includes a symmetric block algorithm implemented in CBC mode with an IV that varies to prevent repeated plaintext patterns. The IV would also prevent replay attacks. As shown in Figure 5.4-2, the IV changes with time, so that aircraft will encrypt legitimate messages with a different IV than the vector used for the replayed squitters. If a deterministic IV is used, such as a counter, the adversary may know the correct IV value at the time of replay. However, the adversary does not know the key (although it is constant during a given KVI) or the plaintext that generated the original squitters, and cannot generate valid ciphertext. This enables the ADS-B processor to use ciphertext as an integrity check as well. As before, aircraft can verify the integrity of DF 19 squitters after receiving at least two squitters that decrypt with identical AA fields.

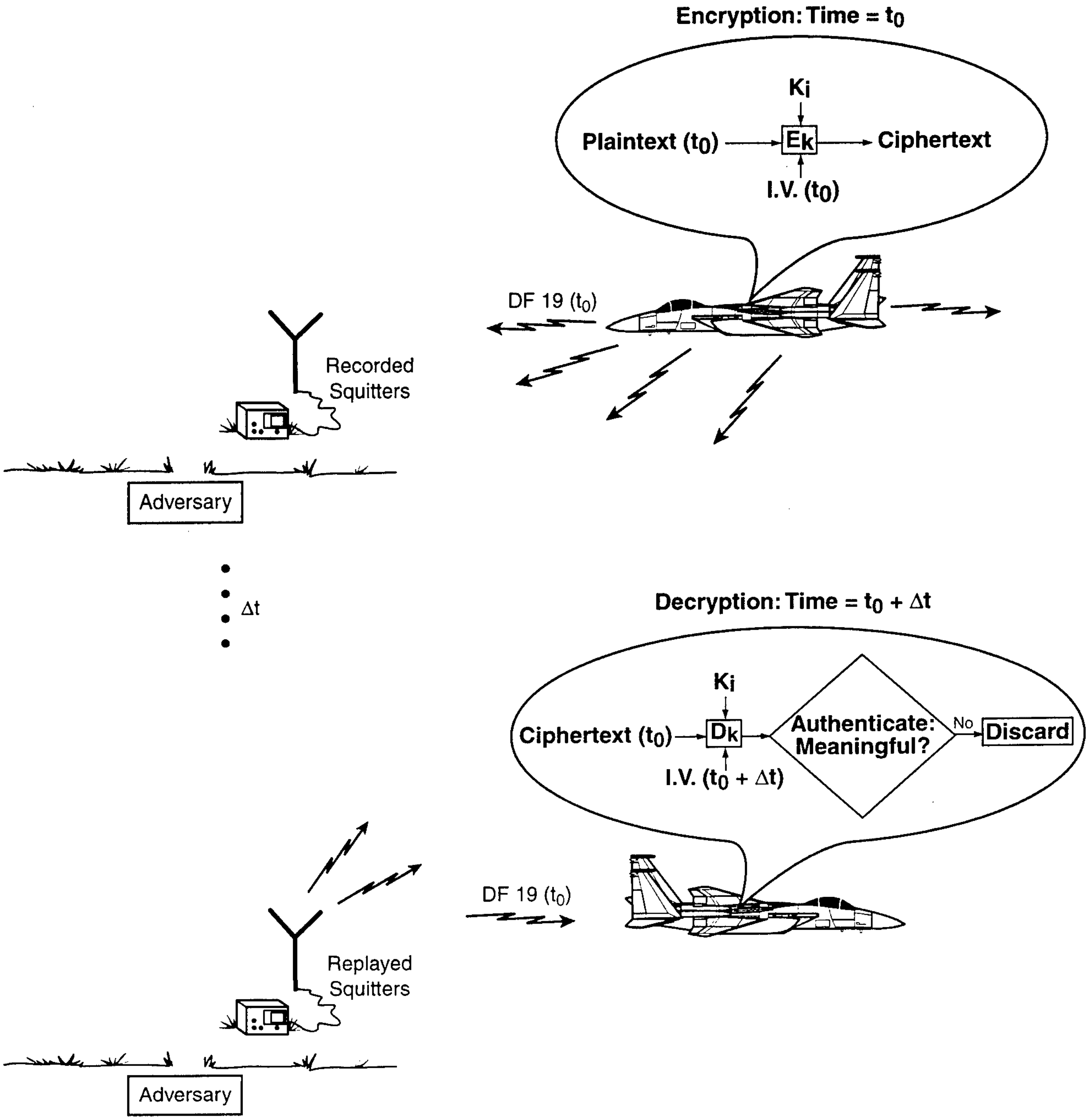


Figure 5.4-2. Use of an integrity check to prevent a replay attack. Valid DF 19 squitters could be implemented in CBC mode with time-varying IVs. After a time delay, a valid squitter will no longer decrypt correctly, as the IV that generated the squitter has expired. The aircraft recognizes the received squitter is not legitimate, and verifies squitter integrity only after receiving two squitters with identical AA fields.

A short KVI also limits an adversary's chance for a successful replay attack. Symmetric keys update after a defined period based on a key update schedule. If the KVI is short, replayed squitters with obsolete keys will decrypt to useless plaintext. The ADS-B processor will disregard any received DF 19 squitter that does not have a match with a previous squitter's AA field.

5.4.2 Checksums and Message Authentication Codes

Use of encryption to authenticate squitters requires analysis of the ME field to recognize intelligible plaintext, or reception of multiple squitters with the same address field. A more robust and simple method to guarantee stand-alone authenticity of each squitter is to include a checksum inside the plaintext. Figure 5.4-3 shows that the checksum adds structure to the plaintext, by appending internal error control to the plaintext before encryption [10]. This method provides authenticity for each message, because it would be difficult for the adversary to generate ciphertext that decrypts with valid error control bits [10].

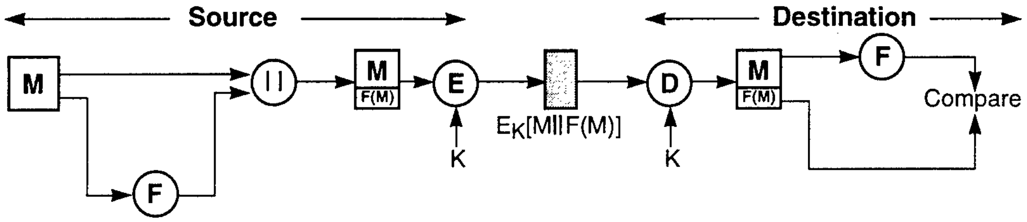


Figure 5.4-3. Internal checksum for authentication. The military aircraft appends a checksum to the plaintext before encryption using the function F, and then broadcasts the squitter. At the destination, the receiver decrypts the message, reproduces the checksum, and authenticates the message if the reproduced checksum matches the transmitted checksum.

This approach requires overhead bits inside the plaintext for the checksum. With further research, DF 19 plaintext may be condensed to permit addition of internal error control. However, this contradicts the assumption in this thesis to utilize existing DF 17 AA and ME fields for DF 19. Therefore, internal checksum authentication is not initially viable.

Other functions regularly used for authentication are message authentication codes, or MACs. MACs are essentially cryptographic checksums, and they require an additional secret key to generate the checksum [10]. A simple symmetric algorithm can implement the MAC, and the operation need not be reversible [11]. That is, the receiver need only reproduce the MAC, and in general cannot reproduce the plaintext given the MAC result because the function is one-way.

One means to implement authentication with a MAC for DF 19 is shown in Figure 5.4-4 [10]. MACs also are problematic for DF 19. The small squitter broadcast size does not lend itself to MACs, as the code cannot overlay the PI field and would significantly impact the message payload. The additional complexity of a secret MAC key is also not desirable.

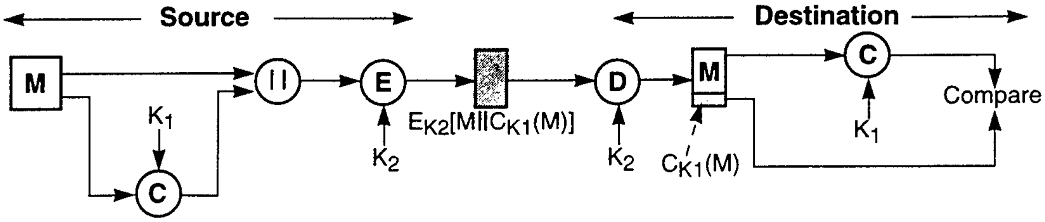


Figure 5.4-4. Message authentication code with confidentiality. In order to both authenticate the message and secure the content of the message, an additional key and encryption step are required.

5.4.3 Summary

Conventional encryption can provide adequate authentication and integrity. Multiple encrypted squitters received from a single source guarantee the messages are legitimate, which combats a masquerade attack. Initialization vectors ensure that a message replayed after some delay is no longer valid which guarantees integrity. In addition, a short KVI deters replay attacks. Should further research enable more compact plaintext fields for DF 19 squitters, a non-cryptographic checksum prior to encryption offers an attractive alternative for ADS-B authentication. This would allow simplified authentication by permitting squitters to stand alone.

5.5 Irregular Block Size

For DF 19 encryption, 64-bit block algorithms were presented in Section 5.3. Processing 80-bit plaintext blocks with a 64-bit algorithm requires non-trivial manipulation and feedback. This section describes a standardized approach called ciphertext stealing for DF 19. Characteristics of publicly-available block algorithms will also be discussed.

Mode S ADS-B squitters are well defined, and the short length of the squitter is suited to broadcast of aircraft state parameters. The original format was not intended for cryptographic purposes. Therefore, less conventional cryptographic methods are needed to overcome the length and format requirements. The data block may be encrypted with conventional symmetric algorithms. However, the format of the squitter is not flexible, so DF 19 encryption must yield a squitter identical in length with no overhead ciphertext bits. Using block algorithms in cipher block chaining (CBC) mode as demonstrated in Section 5.3, each squitter must stand alone and decrypt independently of other broadcasts.

Block algorithms usually have 64-bit block sizes, or multiples thereof. The Data Encryption Standard (DES) has undoubtedly contributed to this pattern, with its 56-bit key and 64-bit block length. DES has been a worldwide encryption standard for over 20 years, and is only now approaching the end of its useful life. The majority of block algorithms bear resemblance to DES with a trend towards longer keylengths to combat more abundant computing resources. A new standard is under development and will be called the Advanced Encryption Standard (AES) [21]. The AES algorithm chosen will offer 128-bit block length and is expected to see widespread use. With the adoption of AES, block algorithm development will most likely continue to be in multiples of 64 bits.

The algorithm chosen for DF 19 must have block length n, such that n is less than or equal to 80 bits. A block length greater than the plaintext block length such as AES is not feasible, as such an approach would require repeated plaintext or padding on each encryption step. The DF 19 algorithm should also be publicly available to ensure that the algorithm has been proven against cryptanalysis. With these considerations, 64 bits is an attractive block size for DF 19 encryption.

With a 64-bit algorithm and CBC, at least two encryption steps are required to encrypt the DF 19 plaintext block. The ciphertext must also be 80 bits long, with no overhead bits. Ciphertext stealing is a method to encrypt a plaintext block given a shorter underlying algorithm, as shown in Figure 5.5-1 [11]. In this method, all plaintext bits go through block encryption, which ensures that plaintext patterns are dispersed in the ciphertext through permutation.

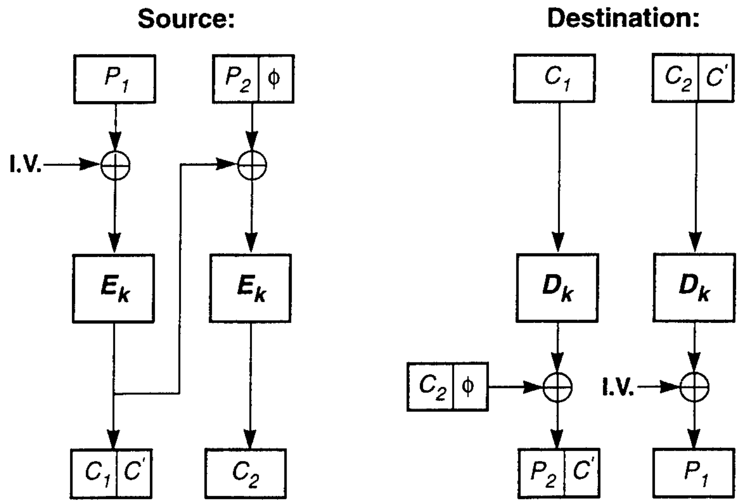


Figure 5.5-1. Ciphertext stealing in CBC mode for DF 19. As with standard CBC, ciphertext is fed back through an exclusive-or with subsequent plaintext. However, with ciphertext stealing, the ciphertext result of the first encryption step $C_1 \| C'$ is exclusive-ored with a short 16-bit plaintext block and then encrypted to generate C_2. The combined 80-bit ciphertext block $C_1 \| C_2$ is broadcast, and the "stolen" ciphertext C' is discarded. The process is reversed at the destination.

The feedback mechanism is the exclusive-or of plaintext with ciphertext from the previous encryption round, which also helps remove patterns and ensure confidentiality. As described in Section 5.3.2, the first block of encryption should include the AA field, which is unique for each aircraft. This helps prevent attacks that take advantage of stereotyped beginnings. The following steps describe the ciphertext stealing process in detail:

(1) Obtain the first 64-bit plaintext block P_1:

$$P_1 = p_1 p_2 p_3 \cdots p_{64}$$

(2) Exclusive-or the time-varying initialization vector IV with P_1

(3) Encrypt the result using the current symmetric key K, generating the short 16-bit ciphertext block C_1 and the temporary ciphertext block C':

$$C_1 \| C' = E_K[P_1 \oplus IV]$$

(4) Zero pad the remaining plaintext bits P_2 to 64-bits, where P_2 is originally

$$P_2 = p_{65} p_{66} p_{67} \cdots p_{80}$$

(5) Exclusive-or the padded plaintext with the ciphertext generated in step (3)

(6) Encrypt the result with the same symmetric key K, generating C_2:

$$C_2 = E_K[(zero - pad(P_2)) \oplus (C_1 \| C')]$$

(7) Obtain the resulting 80-bit ciphertext:

$$C = C_1 \| C_2$$

Note that the "stolen" ciphertext C′ is an intermediate step and is not broadcast. The process is reversed at the receiver, as shown above in Figure 5.5-1.

A proven 64-bit symmetric block algorithm is advantageous for DF 19. The ciphertext stealing with the CBC method illustrated above is just one method to encrypt squitters with 64-bit algorithms. An example of a more complex method is provided in [22]. However, the advantages of ciphertext stealing are its simplicity and efficiency. Simple feedback involves only exclusive-or operations that can be executed in a single clock cycle [23]. The method illustrated is also efficient in that it requires only two encryption steps at the source and destination.

5.6 Symmetric Keys

The primary role of encrypted ADS-B is to provide confidentiality for tactical surveillance. Encryption should ensure that military users may broadcast secure information without risk of eavesdroppers successfully decrypting and exploiting that information. Encryption should also guarantee that received squitters have not been modified. To accomplish these objectives with a block algorithm, the underlying algorithm should satisfy strict avalanche criteria. Sections 5.3 and 5.5 demonstrated that DF 19 could be implemented with cipher block chaining and ciphertext stealing. Care should be taken to remove patterns in the squitter ciphertext, and all DF 19 squitters should be authenticated similarly to methods described in Section 5.4. If these steps are implemented correctly, there should be no shortcut for an adversary to attack DF 19. However, the strength of the encryption system lies in the symmetric keys used by the block algorithm. The keys should be updated frequently, so that each key is valid during a given key validity interval (KVI) or key duration. Essentially, keylength places an upper bound on the security of the system [24]. This section discusses symmetric keys and related issues central to ensuring the confidentiality of DF 19 encryption. Special consideration is given to projected adversaries and their impact on keylength and KVI.

5.6.1 DF 19 Adversaries

An adversary is any unauthorized party hoping to intercept or interfere with confidential communication. Although active attacks are problematic to defend against, an adversary that can intercept and exploit squitters at will is much more dangerous. Because of the open link, such an adversary could obtain information without revealing the success of their attack. The compromised encryption system would provide an endless stream of useful information to the adversary, while military users continue to broadcast information they assume is confidential. Therefore, the strength of the key should be ensured so that the assets required to compromise the key are not economically or physically feasible.

The encrypted squitter adversary is not necessarily well-defined. Consider an aviation enthusiast who is familiar with encryption and has a 1090 MHz receiver. Information on the Joint Strike Fighter obtained by such an adversary and then posted on the Internet is exploitation. In contrast, a 1090 MHz receiver placed by an enemy in the vicinity of a runway is a different kind of adversary. Such a receiver could intercept encrypted squitters of all aircraft departing friendly airspace for combat and forward those squitters to a crypto-computer. Of course, DF 19 encryption should prevent all attacks, but the primary threat is the military adversary. Keylength and KVI must be based upon the assets available to this adversary.

5.6.2 Keylength Estimates

The assets an adversary is willing to allocate for cryptanalysis are related to the value of the encrypted information [11]. This requires speculation about the value of information broadcast for DF 19, as there has been no operational guidance for the use of encrypted squitters in secure environments. On one extreme, an adversary would pay to cover the battlefield with cryptographic hardware and 1090 MHz receivers if they could decrypt squitters and obtain real-time surveillance of all military aircraft. However, in hostile combat situations, all squitter emissions would most likely be reduced in power or completely suppressed. In the absence of operational guidance, encrypted squitter keylength should be valued similarly to other tactical military information. For example, a 1992 estimate recommended 56 to 64-bit keys for tactical military information that must be secure for periods of minutes to hours [11].

The adversary's assets and capabilities must also be projected for the entire lifetime of the system, assumed to be 20 years. A guiding rule-of-thumb for projected assets is Moore's Law, which states that the ratio of processing speed to its cost doubles every 18 months [11].

Alternately, the cost for a given task decreases by approximately a factor of ten every five years. Table 5.6-1 compares the time required to break an n-bit key at a given cost [11]. The table is based on a brute-force machine designed with specialized chips and boards in a massively-parallel configuration. The baseline assumption is the ability to break a 56-bit DES key in 1995 for $1 million in 3.5 hours. The times presented in the table are extrapolated from this baseline 3.5 hour estimate, using Moore's Law. Most industrialized nations' military budgets could afford $1 million to dedicate to cryptanalysis of ADS-B squitters. As shown by the table, while a 64-bit key would secure communications initially, this approach would be insufficient in twenty years.

Table 5.6-1. Present and Projected Key Strength Based on a Brute-Force Attack

	Length of Key in Bits, Years 2000 and 2020					
	64		80		90	
Cost	**2000**	**2020**	**2000**	**2020**	**2000**	**2020**
$1 million	4 days	32 seconds	700 years	24 days	7×10^5 years	70 years
$10 million	9 hours	3 seconds	70 years	2.4 days	7×10^4 years	7 years
$100 million	1 hour	0.3 seconds	7 years	6 hours	7,000 years	255 days
$1 billion	5.4 minutes	0.03 seconds	245 days	36 minutes	700 years	25 days

5.6.3 Assets for a Brute-Force Attack

Assuming there are no underlying vulnerabilities in the chosen algorithm or ciphertext patterns, the adversary must conduct a brute-force attack. The most efficient attacks are achieved when customizable hardware conducts the attack. Two specific alternatives exist [24]. Application specific integrated circuits (ASICs) could be specially designed and customized for a particular cryptanalytic application. This alternative achieves maximum performance, but it is also less attractive because of significant initial investment and less flexibility. However, once designed, mass-produced ASICs are very effective and could test 200 million keys per second for as little as $10 per chip [24]. Complex programmable logic devices (CPLDs) provide greater flexibility through reconfigurable logic, also at very attractive prices. Massively-parallel collections of CPLDs can be used to perform an exhaustive key search, as the keyspace can be broken down into independent searches without penalty [25]. A CPLD brute-force attack against

DF 19 appears more feasible than the ASIC approach, as the design costs for an ASIC probably exceed the value of DF 19 information, even for government intelligence agencies.

In general, a brute-force search can be modeled after "known-plaintext" cryptanalysis, where the adversary knows some part of the original message plaintext. After testing a trial key, the adversary would compare the known plaintext to the decrypted message. For ADS-B, an adversary could use the address field as known plaintext. For example, the adversary could decrypt squitters attempting to find squitters that originate from the same aircraft by comparing the address announced (AA) fields in the decrypted messages. If the AA plaintext segments do not match, the key is incorrect and should be discarded.

The cost of the attack is estimated based upon the cost to conduct an attack in some maximum amount of time. The number of CPLD chips N required to conduct the attack can be estimated by

$$N = \frac{2^{n-1}(CPK)(CS)}{t_{req}}$$

where n is the keylength in bits, CPK is the number of clock cycles required to set-up and test each key, CS is the clock speed of each chip, and t_{req} is the maximum average time to conduct the search [25]. Note that the average time requires search of one-half of the keyspace on average. Equivalently,

$$N = \frac{2^{n-1}/r}{t_{req}}$$

where r is the key test rate in keys per unit time as in equation (3.3-1). Then the estimated cost of the search is

$$Cost = N \cdot p$$

where p is the price per CPLD chip.

For example, consider use of DES to encrypt DF 19 squitters with CBC and ciphertext stealing, where the adversary has knowledge of the deterministic initialization vector. Assume a simplified scenario where an adversary obtains two same-type squitters from a single aircraft, and the adversary hopes to break the key in at least one year's time. Under these assumptions, a brute force attack would require approximately 90 clock cycles per key to set up each trial key, decrypt two messages, and compare the AA fields [25]. Given a CPLD that operates at

200 MHz, the brute-force attack described above that averages one year to complete would require approximately 550 CPLD chips. If each chip costs less than $100, the attack would be economical for military adversaries.

5.6.4 Security for the lifetime of DF 19

As illustrated by this hypothetical attack, the tactical nature of DF 19 information is central to the problem. Again, tactical confidentiality of squitters is the primary objective for encryption. For a military adversary hoping to obtain real-time surveillance information, the ability to compromise keys in one year is not of any tactical value. For such a near-term brute force attack, an 80-bit key is sufficient for at least 20 years. An adversary willing to spend $10 million would still only recover the key after a couple days. This type of attack to obtain tactical information would not be economically feasible for an adversary.

A far-term brute force attack is also worth considering, although this attack is not the primary security objective for DF 19. Again, returning to the Joint Strike Fighter example, consider an adversary hoping to obtain information about the aircraft. For example, the adversary could attempt to intercept secret information about its flight characteristics by decrypting squitters. This information is not tactical, and an adversary may be willing to spend $1 million to gain insight into training and capabilities of U.S. aircraft, even if it requires 24 days or more to generate this information. Therefore, an 80-bit key may be insufficient for far-term confidentiality.

In 1996, a group of cryptographers and computer scientists released a report with keylength recommendations for symmetric algorithms. They emphasized that good cryptographic practice calls for longer keylength than might appear to be required [24]. This is because many believe that keylength is not an absolute measure of security due to the difficulty of removing all cryptanalytic shortcuts. For military ADS-B, a prudent approach may be to recommend keylength based on far-term confidentiality and to include extra bits for an additional safety factor, despite a primary objective for tactical confidentiality. The same group of computer scientists recommended at least 90-bit keylength for systems expected to be utilized for the next 20 years [24]. Although an 80-bit key appears sufficient for tactical information, a keylength of at least 90 bits should be sufficient for encrypted squitters. This satisfies tactical confidentiality objectives and renders brute-force attacks infeasible. A 90-bit keylength should also protect DF 19 squitters from prolonged brute-force attacks for the lifetime of the system, assuming an adversary is not willing to allocate $10 million for DF 19 cryptanalysis.

5.6.5 Key Validity Interval

As with all cryptosystems, keys expire. The lifetime of each symmetric key is defined as the key validity interval. The KVI does not explicitly increase the security of individual squitters against brute-force attacks. However, the KVI does limit the amount of information lost to a successful attack if the keys are not related, as the adversary will have to conduct another attack for each symmetric key update. This greatly discourages a short-term brute-force attack, as the adversary has little incentive to break a key when the keys change rapidly. Although the exact length of the KVI is arbitrary, a short length is desirable. For example, a compromised symmetric key with a KVI on the order of several seconds will reveal only a short window of surveillance information. As illustrated in Section 5.4, a short KVI also aids integrity by preventing playback attacks. A message replayed with an expired key will decrypt to useless information and will be discarded by the ADS-B processor.

KVI is a concept familiar to tactical military systems, most notably identification of friend or foe (IFF). For IFF systems, the KVI is often classified. The keys update at regular intervals, enabling secure combat identification. A similar key management approach should be taken for DF 19. The keys should be uploaded onto military aircraft, and a key schedule based on predefined KVIs should update symmetric keys sequentially on all aircraft using synchronized GPS time. Memory allocation is not a major consideration for keys, as daily upload of 90-bit keys would require very little memory space, even with very short KVIs.

5.6.6 Summary

A keylength of at least 90 bits appears realistic for DF 19. This keylength provides a level of confidentiality that will deter brute-force attacks and ensure the confidentiality of DF 19 squitters for up to 20 years in the absence of revolutionary advances in cryptography. The key should update frequently. A short KVI will also deter brute-force attacks by limiting the information gained by a successful attack, and will also aid integrity.

6. Conclusions and Recommended Research

In this chapter, the results of previous chapters are summarized, and a candidate encryption system for Mode S ADS-B is presented. This chapter concludes that a feasible encryption system does exist to provide tactical confidentiality for squitters, and also that additional research on encrypted Mode S ADS-B could result in a system that helps prevent mid-air collisions.

6.1 Recommendations for a Feasible Encryption System

6.1.1 System Recommendations

Tactical military aircraft are equipped with transponders for ground surveillance and IFF. This thesis assumes that these aircraft will soon equip with Mode S-capable transponders, in accordance with equipment mandates and transponder upgrade schedules. The following are additional subsystems required for encryption and ADS-B:

- 1090 MHz Receiver: The receiver should receive, demodulate, and digitize incoming squitters. In addition, the receiver should perform error detection and correction on the squitters before passing the data to the ADS-B processor. The receiver should share existing antennas used by the transponder.
- Timing System: The timing system should include a transponder clock that can synchronize with the GPS PVT updates, to allow synchronized squittering. This system also enables position extrapolation, so that aircraft can broadcast more accurate representations of their instantaneous position.
- ADS-B Processor: The processor should interpret all incoming squitters, both encrypted and non-encrypted, as well as access and format data for outgoing squitters. It serves as the primary interface to navigation and display systems external to the transponder. The processor should also interface with the ADS-B crypto unit.
- ADS-B Crypto Unit: The crypto unit should decrypt incoming encrypted squitter ciphertext and encrypt outgoing squitter plaintext. In addition to this primary role, the unit should store symmetric keys for the encryption algorithm and perform automated updates of the keys based on a prearranged key validity interval. The crypto unit should be a controlled cryptographic item, and should be physically

separable from the transponder to permit ease of upgrade and repair as well as simplified security procedures.

A Mode S ADS-B encryption system would also interface in new ways with existing navigation and display systems external to the transponder. GPS should serve as the primary navigation source, and ADS-B information should be incorporated into existing aircraft displays to help increase situational awareness.

6.1.2 DF 19 Format Issues

For practical considerations such as efficient upgrade to DF 19 capability and for ease of analysis, this thesis assumed that DF 19 should initially mimic DF 17, the civil extended squitter. Under this assumption, the existing error detection and correction scheme should be utilized, which allows robust error correction in an environment dominated by burst errors. This scheme requires 24 parity bits, occupying the 24 least significant bits of the DF 19 squitter. The message content of the encrypted squitter plaintext should initially include the DF 17 address announced (AA) and message (ME) fields, which occupy 80 bits. ME fields are well-defined for several squitter subtypes including position and velocity types required for situational awareness. It is noted that further research should allow consolidation of bits within the AA+ME field or development of entirely new data content, due to the greater standardization of military hardware. Such consolidation should permit more efficient data broadcast as well as more robust encryption services such as internal checksums for authentication. At a minimum, this 80-bit AA+ME field should be encrypted, as both the address and surveillance data must remain confidential. The parity bits must be broadcast in the clear, without encryption.

A significant difference between the DF 17 and DF 19 squitter formats is the need for an application field (AF). DF 19 is reserved for use as a military squitter. However, different applications may be developed for DF 19, and not all applications need be encrypted. As such, the application of each squitter must be explicit through the use of control bits broadcast in the clear. Initial guidance by ICAO calls for eight control bits at the eight most significant bits of the broadcast. The first five bits are the standard Downlink Format and are assigned 19. The remaining three bits should be the AF, which replace the transponder capability (CA) bit field used by DF 17 squitters. This allows eight different military applications for DF 19. This approach is feasible, given that eight applications are sufficient and that the CA field is not necessary due to military equipment standardization. The notation used in this thesis calls for an application field set equal to one for the encrypted squitter, such that DF = 19 and AF = 1. The resulting squitter format for encrypted DF 19 is shown in Figure 6.1-1. The squitter has eight control bits, 80 ciphertext bits, and 24 parity bits.

(5)	(3)	(80)	(24)
DF 19	AF 1	*Ciphertext Field*	PI

Figure 6.1-1. Recommended format for DF 19. The italicized Ciphertext Field indicates that the 80-bit segment is encrypted.

6.1.3 Encryption Feasibility and Recommendations

The military ADS-B environment is susceptible to attacks by adversaries attempting to access or corrupt squitter information. Attacks are broadly characterized as passive or active, and more specific types of attacks applicable to ADS-B are categorized as interception, modification, and fabrication. Chapter 5 discussed specific examples of these attacks and methods to overcome them. The attacks and corresponding defensive measures to counter the attacks are summarized by Figure 6.1-2.

The objective of the thesis was to determine the feasibility of encryption for the Mode S datalink. As discussed in Chapter 5, solutions to ADS-B attacks are available, and encryption is feasible. This conclusion is best summarized by reviewing the security services that are considered desirable for encrypted squittering: confidentiality, identification, authentication, and integrity.

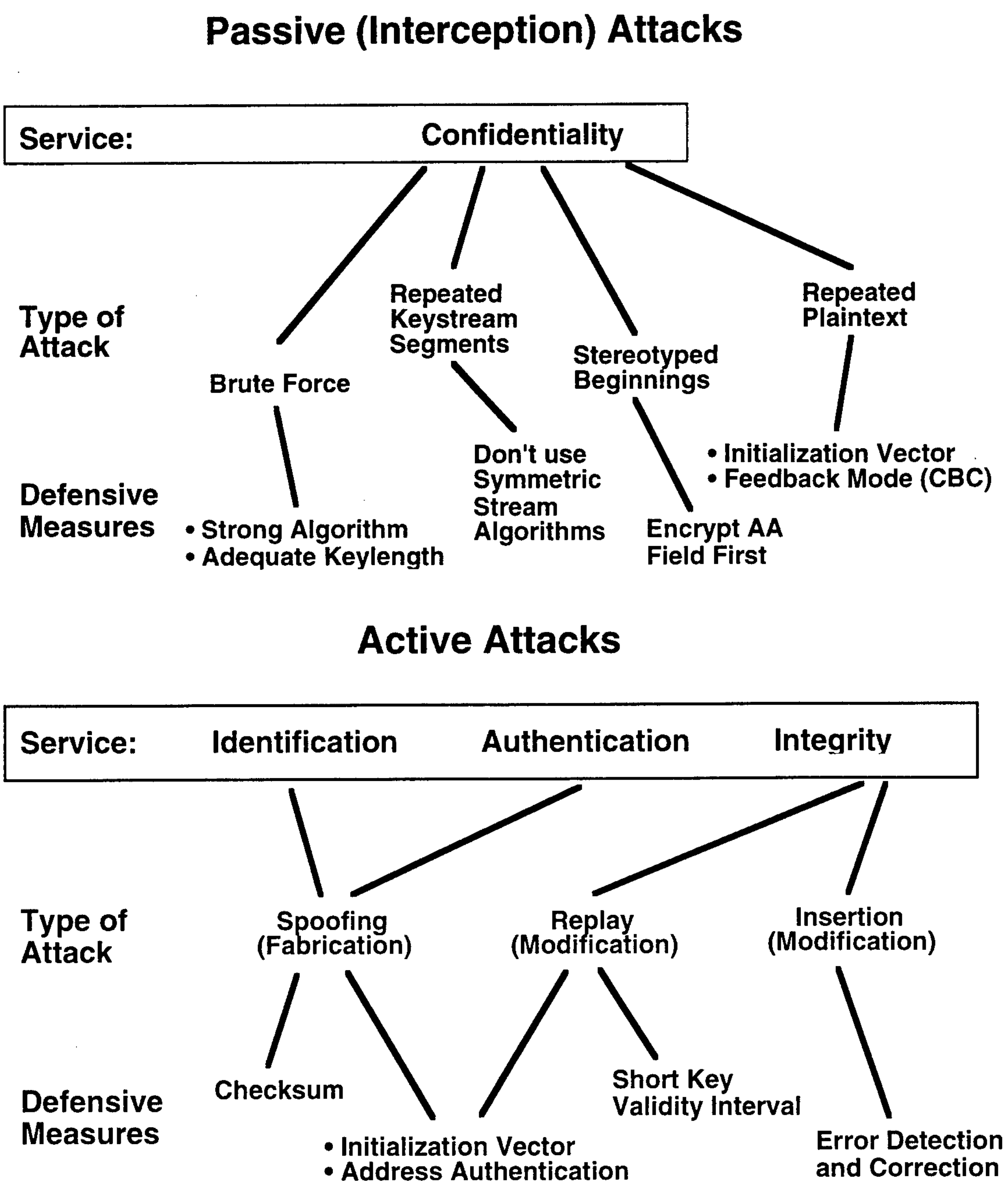


Figure 6.1-2. Summary of DF 19 attacks and methods to counter those attacks, as discussed initially in Chapter 5. Attacks are divided into passive and active attacks. Attacks are then further divided into specific types of attacks. Each attack aims to exploit vulnerabilities in a required security service. The security service can be strengthed by a defensive measure to help counter a specific attack.

6.1.3.1 Confidentiality

The essential service for military ADS-B is tactical confidentiality. This is due to the fact that an adversary could intercept and utilize information without revealing the success of the attack, essentially capturing an endless source of real-time surveillance information. Pilots should be guaranteed that information broadcast cannot be intercepted by an adversary and exploited while the information is still useful. This guarantee is possible by first ensuring that there are no vulnerabilities in the algorithm that would simplify cryptanalysis. Secondly, the complexity of an attack should be so great that the assets required to successfully complete such an attack are either not available or greatly exceed the value of the information to be gained.

As shown in Figure 6.1-2, a brute-force interception attack is one attack against confidentiality. This attack does not attempt to take advantage of any weakness in the cryptosystem, but instead tests every key until the information is compromised. There are several methods to stifle this attack. First and foremost, the symmetric algorithm chosen for the cryptosystem should be proven to have no known weaknesses, so that the adversary cannot simplify the brute-force attack. A publicly-available algorithm proven to be strong by withstanding scrutiny from the hacker community is recommended. Cryptanalysis results for algorithms with known weaknesses are generally available on the Internet. Likewise, the strength of other algorithms is also represented. A variety of contemporary public 64-bit block algorithms are worth consideration for DF 19 [26, 27]: RC5 has variable block length and variable keylength. Blowfish also has variable keylength, but may be less attractive for DF 19, because the algorithm requires a fairly large key set-up time that limits its utility for applications that change keys frequently [23]. Skipjack is a block algorithm developed by the NSA, but it has an 80 bit key which may be insufficient for DF 19. The algorithm was initially secret, but has recently been released to the public. Two patented algorithms are also worth consideration. IDEA is considered very strong, and it has a 128-bit key. REDOC II, in contrast with the above algorithms, has an 80-bit block length, but cryptanalysis results are less available due to its obscurity. It uses a 160-bit key.

If a novel algorithm is designed for DF 19, the algorithm should be tested thoroughly. The chosen or designed algorithm should satisfy strict avalanche criteria (SAC) in S-box design, so that if only one bit in the input to the S-box changes, at least half of the corresponding output bits change. This ensures non-linearities in the algorithm by spreading bit dependencies, which makes the system more resistant to cryptanalysis.

Given a strong algorithm, the complexity of the attack is determined by the keylength. For each additional bit, the expected time to complete a brute-force attack is doubled. Correspondingly, the time and computer assets required to complete a successful attack increases with each bit. The recommended keylength for Mode S DF 19 is at least 90 bits. This length should ensure tactical confidentiality, and also provide long-term confidentiality for approximately 20 years.

An additional type of passive interception attack occurs when the adversary hopes to exploit the squitter format to simplify cryptanalysis. One manifestation of this type of attack could arise due to repeated keystream segments for stream algorithms. Due to the broadcast nature of squittering, a method to synchronize keystreams could involve segmenting the keystream in to 80-bit segments. Although segmentation of the keystream is poor technique for stream applications, the greater damage is that this approach would create a pattern that essentially removes the keystream from cryptanalysis. That is, an adversary could gather information without compromising the key or the keystream. For this reason, symmetric stream algorithms are not desirable for DF 19.

Another interception attack that could permit exploitation of the squitter format is the stereotyped-beginnings attack. In this attack, an adversary could attempt to simplify cryptanalysis by recognizing that segments of the ME field inside the plaintext are constant for squitter subtypes. This may create patterns in the ciphertext that an adversary could exploit. This attack can be countered by controlling the order that the data is encrypted, and by ensuring that S-box design satisfies SAC. Under the assumption that DF 19 utilizes the DF 17 AA+ME plaintext, the first encryption step should include the AA field. Each aircraft Mode S address is unique, which guarantees that at least one bit varies between aircraft for the inputs to the encryption algorithm. With this approach, squitters from different aircraft would be dissimilar, despite similar header information in the ME field and the use of identical symmetric keys.

A final interception attack worrisome for DF 19 is a repeated plaintext attack. As opposed to stereotyped beginnings, this attack would attempt to exploit same-subtype squitters broadcast by a single aircraft. For example, if the first plaintext block input to the encryption algorithm does not vary from squitter to squitter, the ciphertext will be identical for those squitters. This vulnerability is exacerbated when none of the plaintext bits vary, which is possible (although unlikely) for a hovering aircraft or an aircraft travelling at a constant velocity vector. In this type of attack, an adversary could attempt to gain information without decrypting the squitter, simply by noticing constant segments in ciphertext. It should be noted that this attack is less problematic, because tactical aircraft flight is dynamic. In general, the surveillance plaintext would be constant for only a short period of time, and the information gained by the adversary would be only marginally beneficial. Nevertheless, this attack can be countered by feedback and use of an initialization vector (IV), or initial state, for the encryption system. With this approach, the IV should change at least as often as the minimum time duration between same type squitter broadcasts. The IV need not be random or secret, but must be synchronized between aircraft. Again, using an algorithm with SAC S-box design is essential, as a single bit difference in plaintext input will create dissimilar ciphertext. The use of IVs may be considered optional, as they require additional data storage and synchronization complexity for an increase in security against an unlikely attack. However, IVs are also beneficial for authentication and integrity, as will be reviewed in Sections 6.1.3.3 and 6.1.3.4.

Finally, a feedback mode such as cipher block chaining helps protect the confidentiality of the algorithm against patterns by dispersing bit dependencies through multiple rounds of encryption. One viable method to incorporate feedback and an IV with a 64-bit symmetric block algorithm is cipher block chaining with ciphertext stealing. With the 80-bit ciphertext block, ciphertext stealing should remove patterns with minimal impact on efficiency, requiring only two encryption and decryption steps per squitter.

6.1.3.2 Identification

Identification is straightforward to provide, as the recommended approach is to mimic DF 17 and include the Mode S address in the AA field as part of the plaintext data. As a result, the address data is encrypted. Once a squitter is received, if the ciphertext is decrypted correctly, own aircraft can identify the source of the squitter through the AA field. Squitters received with error should be error corrected or discarded, allowing the corrected squitter to be correctly decrypted and identified. Fabricated squitters (spoofing) with deceptive identification will be protected against by use of initialization vectors and address authentication.

6.1.3.3 Authentication

As was shown in Figure 6.1-2, a spoofing attack is an active fabrication attack, where an adversary generates squitters in an attempt to deceive or corrupt the ADS-B system. Spoofing attacks also threaten the identification service with false aircraft identification, as described in the previous paragraph. To combat this attack, the ADS-B processor must differentiate legitimate squitters from spoofed squitters that appear to be legitimate. This service is known as authentication and would prevent fabricated squitters from bogging down the ADS-B processor and also prevent the display of misleading information to pilots.

Methods discussed to provide authentication and prevent spoofing attacks include basic encryption, internal checksums, and message authentication codes or MACs. MACs are not feasible for DF 19, because they require overhead bits beyond the capacity of the 80-bit ciphertext block. In addition, use of MACs would require additional complexity and additional encryption keys.

Encryption for ADS-B should provide adequate authentication, under the notion that only authorized users have access to the symmetric keys. As such, only authorized users should be able to decrypt ciphertext into useful information. With this approach, the structured format of the squitter provides a benefit, because the formatting can be used to differentiate authentic squitters from fabricated squitters. The suggested approach is address authentication, where own aircraft authenticates squitters only after receiving at least two squitters from a single source (i.e., at least two squitters with identical AA fields). The use of an initialization vector is also beneficial for authentication, because an adversary could not repeatedly broadcast a spoofed squitter that would decrypt with the same aircraft identification. Such an attempt would decrypt to a different aircraft identity with each IV update.

Internal checksums are an attractive defensive measure for authentication, as each squitter could be authenticated independently. However, this approach is not initially recommended, as the checksum would require overhead bits inside the squitter plaintext.

6.1.3.4 Integrity

As with the civil squitter, parity encoding should provide a level of integrity for DF 19. Encoding should occur after encryption at the squitter source, and decoding should occur before decryption at the squitter destination. Mode S error detection and correction would pass only corrected squitters to the ADS-B processor for processing and decryption. Squitters that are not correctable would be discarded. This should remove the impact of interference such as synchronous garbling of ATCRBS and Mode S replies and multipath. Mode S EDC would also protect against insertion attacks (which are a very unlikely and futile attack for DF 19), where an adversary would intentionally insert burst energy into the squitter environment to disrupt services. However, additional cryptographic methods to check squitter integrity are required for other types of active attacks.

The primary attack against integrity discussed in Chapter 5 was the replay attack. In this attack, the adversary records DF 19 squitters and replays them after some time delay to provide misleading information to authorized DF 19 users. Once again, conventional encryption can be used to verify squitter integrity and prevent replay attacks. More specifically, IV and short KVIs should be implemented as additional features to the conventional encryption algorithm to prevent replay attacks. A valid squitter that is recorded by an adversary will no longer be valid once the IV used to generate that squitter expires. Even if the adversary has knowledge of the IV at the time of replay, the adversary does not have access to the key or original data used to generate the squitter. The squitter will decrypt to useless information. As a result, a similar mechanism such as address authentication used to authenticate squitters can also verify squitter integrity. Once at least two squitters with identical AA fields are received, the ADS-B processor should be satisfied with squitter authenticity and integrity and should begin tracking that aircraft.

A short KVI provides a similar integrity service. A replayed squitter that was generated with an expired symmetric key will decrypt to meaningless information and should be discarded.

6.1.4 Conclusions

A Mode S ADS-B encrypted squitter is feasible. This thesis shows that conventional encryption methods can be used, and furthermore these methods provide the required security services for DF 19: tactical confidentiality, authentication, and integrity. Techniques can be implemented using existing extended squitter formatting, parity encoding, and data content, with only minor modifications for an application field. The approach illustrated calls for a publicly-available 64-bit block algorithm, and a 90-bit or greater symmetric key. The key should be updated frequently. An initialization vector can be used to prevent ciphertext patterns and to aid authentication and squitter integrity. Cipher block chaining should also be used to help conceal patterns. Ciphertext stealing can be used to encrypt the 80-bit block with a 64-bit algorithm.

6.2 Suggested Research

Although other candidate methods were discussed, this thesis provided only one approach for a feasible cryptosystem for DF 19. The approach presented here illustrates that encryption for ADS-B is feasible, but this approach should not be considered limiting. Rather, the discussion of squitter vulnerabilities and attacks should provide a foundation for further research. As discussed throughout the thesis, military hardware standardization should allow bit consolidation inside the squitter message payload. For example, a shortened aircraft address could replace the 24-bit Mode S address and still provide identification for authentication and integrity. Efficient bit use could result in a combined position / velocity DF 19 squitter. Military users should consider opportunities such as these to best utilize DF 19 for their situational awareness needs.

Research also should be conducted on the ADS-B processor architecture and memory requirements. The DF 19 squitter environment should be modeled, and the time required to decrypt and process incoming squitters must be analyzed.

A separate operational area of research would concern key distribution and key upload onto aircraft. If symmetric keys are compromised before they reach the aircraft, the strength of the encryption system is irrelevant. Secure means to distribute the keys must be established. Potential solutions may include public-key encryption.

Actual operational implementation of the encrypted squitters will also require further study. Guidance to pilots for use of encrypted DF 19 must be explicit. For example, military aircraft who broadcast DF 11 (short squitters) or DF 17 squitters before switching to encrypted mode may inadvertently provide their aircraft identification to adversaries. The adversary could then use the intercepted addresses to simplify cryptanalysis of DF 19 squitters as the aircraft switch to encrypted mode in hostile airspace. Actual key validity interval is another example of the operational issues that must be considered.

Appendix A. DF 17 Format Type Codes [14]

Format Type Subfield Code Definitions (DF = 17)				
Type Code	**Format Type**	**Horizontal Protection Limit, HPL**	**95% Containment Radius, μ and ν, On Horizontal and Vertical Position Error**	**Altitude Type**
0	No Position Information			Baro Altitude or No Altitude Information
1	Identification (Category Set D)			*Not Applicable*
2	Identification (Category Set C)			*Not Applicable*
3	Identification (Category Set B)			*Not Applicable*
4	Identification (Category Set A)			*Not Applicable*
5	Surface Position	HPL < 7.5 m	$\mu < 3$ m	No Altitude Information
6	Surface Position	HPL < 25 m	3 m $\le \mu < 10$ m	No Altitude Information
7	Surface Position	HPL < 185.2 m (0.1 nm)	10 m $\le \mu < 92.6$ m (0.05 nm)	No Altitude Information
8	Surface Position	HPL > 185.2 m (0.1 nm)	(0.05 nm) 92.6 m $\le \mu$	No Altitude Information
9	Airborne Position	HPL < 7.5 m	$\mu < 3$ m	Baro Altitude
10	Airborne Position	7.5 m $\le$ HPL < 25 m	3 m $\le \mu < 10$ m	Baro Altitude
11	Airborne Position	25 m $\le$ HPL < 185.2 m (0.1 nm)	10 m $\le \mu < 92.6$ m (0.05 nm)	Baro Altitude
12	Airborne Position	185.2 m (0.1 nm) $\le$ HPL < 370.4 m (0.2 nm)	92.6 m (0.05 nm) $\le \mu <$ 185.2 m (0.1 nm)	Baro Altitude
13	Airborne Position	370.4 m (0.2 nm) $\le$ HPL < 926 m (0.5 nm)	185.2 m (0.1 nm) $\le \mu <$ 463 m (0.25 nm)	Baro Altitude
14	Airborne Position	926 m (0.5 nm) $\le$ HPL < 1852 m (1.0 nm)	463 m (0.25 nm) $\le \mu <$ 926 m (0.5 nm)	Baro Altitude
15	Airborne Position	1852 m (1.0 nm) $\le$ HPL < 3704 m (2.0 nm)	926 m (0.5 nm) $\le \mu <$ 1.852 km (1.0 nm)	Baro Altitude
16	Airborne Position	3.704 km (2.0 nm) $\le$ HPL < 18.52 km (10 nm)	1.852 km (1.0 nm) $\le \mu <$ 9.26 km (5.0 nm)	Baro Altitude
17	Airborne Position	18.52 km (10 nm) $\le$ HPL < 37.04 km (20 nm)	9.26 km (5.0 nm) $\le \mu <$ 18.52 km (10.0 nm)	Baro Altitude
18	Airborne Position	HPL > 37.04 km (20 nm)	18.52 km (10.0 nm) $\le \mu$	Baro Altitude
19	Airborne Velocity	*Not Applicable*	*Not Applicable*	*Difference between "Baro Altitude" and "GNSS Height (HAE) or GNSS Alt (MSL)"*
20	Airborne Position	HPL < 7.5 m	$\mu < 3$ m and $\nu < 4$ m	GNSS Height (HAE)
21	Airborne Position	HPL < 25 m	$\mu < 10$ m and $\nu < 15$ m	GNSS Height (HAE)
22	Airborne Position	HPL $\ge$ 25 m	$\mu \ge 10$ m or $\nu \ge 15$ m	GNSS Height (HAE)
23	Reserved for Test Purposes			
24	Reserved for surface system status			
25 - 27	Reserved			
28	Extended Squitter Aircraft Status			
29	Current/Next Trajectory Change Point			
30	Aircraft Operational Coordination			
31	Aircraft Operational Status			

Appendix A. DF 17 ME field: Position Squitter [14]

1	
2	FORMAT TYPE CODE
3	
4	
5	
6	SURVEILLANCE STATUS
7	
8	SINGLE ANTENNA FLAG (SAF)
9	
10	
11	
12	ALTITUDE (barometric or GPS HAE, as specified by the format type code)
13	
14	
15	
16	
17	
18	
19	
20	
21	TIME SYNCHRONIZATION (T)
22	CPR FORMAT (F)
23	MSB
24	
25	
26	
27	
28	
29	ENCODED LATITUDE
30	(CPR airborne format)
31	
32	
33	
34	
35	
36	
37	
38	
39	LSB
40	MSB
41	
42	
43	
44	
45	
46	ENCODED LONGITUDE
47	(CPR airborne format)
48	
49	
50	
51	
52	
53	
54	
55	
56	LSB

PURPOSE : To provide accurate airborne position information.

Surveillance status shall be coded as follows:

0 = No condition information

1 = Permanent alert (emergency condition)

2 = Temporary alert (change in Mode A identity code other than emergency condition)

3 = SPI condition

Codes 1 and 2 shall take precedence over code 3.

SAF set to "0" indicates a single transmit antenna. SAF set to "1" indicates a dual transmit antenna system.

F equal to "0" denotes even format coding, and F equal to "1" denotes odd format coding.

T equal to "0" indicates the position report time of applicability is not synchronized to GPS time. T equal to "1" indicates that the time of applicability is synchronized to GPS time.

Appendix A. DF 17 ME field: Airborne Velocity [14]

1	MSB			1
2	0			0
3	FORMAT TYPE CODE = 19			0
4				1
5	LSB			1
6	SUBTYPE 1	0	SUBTYPE 2	0
7		0		1
8		1		0
9	INTENT CHANGE FLAG			
10	IFR CAPABILITY FLAG			
11	NAVIGATION UNCERTAINTY			
12	CATEGORY – VELOCITY			
13	(NUC_R)			
14	DIRECTION BIT for E-W velocity: 0 = East, 1 = West			
15	EAST-WEST VELOCITY			
16	NORMAL: LSB = 1 knot		SUPERSONIC: LSB =4 knot	
17	All zeroes = no velocity info		All zeroes = no velocity info	
18	Value	Velocity	Value	Velocity
19	1	0 kt	1	0 kt
20	2	1 kt	2	4 kt
21	3	2 kt	3	8 kt
22	-	-	-	-
23	1022	1021 kt	1022	4084 kt
24	1023	>1021.5 kt	1023	>4086 kt
25	DIRECTION BIT for N-S Velocity, 0=North, 1 = South			
26	NORTH-SOUTH VELOCITY			
27	NORMAL : LSB = 1 knot		SUPERSONIC : LSB =4 knot	
28	All zeroes = no velocity info		All zeroes = no velocity info	
29	Value	Velocity	Value	Velocity
30	1	0 kt	1	0 kt
31	2	1 kt	2	4 kt
32	3	2 kt	3	8 kt
33	-	-	-	-
34	1022	1021 kt	1022	4084 kt
35	1023	>1021.5 kt	1023	>4086 kt
36	SOURCE BIT for vertical rate: 0 = GNSS, 1 = Baro			
37	SIGN BIT for vertical rate: 0 = Up, 1 = Down			
38	VERTICAL RATE			
39	All zeroes - no vertical rate information, LSB = 64 ft/min			
40	Value	Vertical rate	Ref. ARINC labels	
41	1	0 ft/min	GPS: 165	
42	2	64 ft/min	INS: 365	
43		-		
44	510	32576 ft/min		
45	511	>32608 ft/min		
46				
47	RESERVED FOR TURN INDICATOR			
48				
49	GNSS ALT. SIGN BIT: 0 = Above baro alt, 1 = Below baro alt			
50	GNSS ALT. DIFFERENCE FROM BARO. ALT.			
51	All zeroes = no info; LSB = 25 ft			
52	Value	Difference		
53	1	0ft		
54	2	25 ft		
55	-	-		
56	126 127	3125 ft >3137.5 ft		

PURPOSE: To provide additional state information for both normal and supersonic flight.

Subtype shall be coded as follows:

Code	Velocity	Type
0	As in first edition of Doc 9688	
1	Ground Speed	Normal
2		Supersonic
3	Airspeed, Heading	Normal
4		Supersonic
5	Reserved	
6	Reserved	
7	Reserved	

IFR capability shall be coded as follows:

0 = Transmitting aircraft has no capability for ADS-B based conflict avoidance or higher level applications.

1 = Transmitting aircraft has capability for ADS-B based conflict avoidance and higher level applications.

ARINC labels for velocity:

East - West	North - South
GPS : 174	GPS : 166
INS : 367	INS : 366

ARINC labels for GNSS source:

GNSS altitude (MSL): 076

GNSS height (HAE): 370

NUC shall be coded as follows:

NUC_R	Horizontal Velocity Error (95%)	Vertical Velocity Error (95%)
0	Unknown	Unknown
1	<10 m/s	<15.2m/s (50fps)
2	<3 m/s	<4.6 m/s (15fps)
3	<1 m/s	<1.5 m/s (5fps)
4	<0.3 m/s	<0.46m/s (1.5fps)

Bibliography

[1] "Formation Flight Mishap Analysis Report," Naval Air Systems Command, 1 October 1998.

[2] V. Orlando, "Mode S Beacon System: A Functional Overview," Lincoln Laboratory Project Report ATC-150, 29 August 1989.

[3] M. Kayton and W. Fried, *Avionics Navigation Systems*, John Wiley & Sons: New York, 1997.

[4] V. Orlando, "ADS-Mode S: Initial System Description," Lincoln Laboratory Project Report ATC-200, 2 April 1993.

[5] J. Welch and V. Orlando, "Traffic Alert and Collision Avoidance System (TCAS): A Functional Overview of Minimum TCAS II," MIT Lincoln Laboratory Group 92 Project Report ATC-119, 8 Apr 1983.

[6] Maj. E. Coyl, USMC UH-1 pilot.

[7] LCDR C. Holladay, USN F-14 pilot.

[8] D. Jonathan Bernays, S. Thompson, and W. Harman, "Measurement of ADS-B Extended Squitter Performance in the Los Angeles Basin Region," 19th AIAA Digital Avionics System Conference, October 2000.

[9] "Measurement of 1090 MHz Extended Squitter Performance and the 1030/1090 MHz Environment in Frankfurt, Germany," Interim Report, FAA/DFS/Eurocontrol, 3 November 2000.

[10] W. Stallings, *Cryptography and Network Security, Principles and Practice,* Prentice Hall, Inc. 1999.

[11] B. Schneier, *Applied Cryptography, Protocols, Algorithms, and Source Code in C*, John Wiley & Sons, Inc. 1996.

[12] T. Goblick, V. Heinz, S. Kolek, K. Saunders, P. Misra, D. Spencer, and A. Lind, "Feasibility Study of the use of Mode S Extended Squitter for the Mid-Air Collision Avoidance System (MCAS)," MIT Lincoln Laboratory Group 92, Memorandum: 92PM-MCAS-0003, 27 Nov 2000.

[13] CI-MAGR-400, *Performance Specification for the Navstar Global Positioning System (GPS) miniature Airborne GPS Receiver 2000 (MAGR 2000).*

[14] ICAO Document 9688, *Manual on Mode S Specific Services*, June 2000.

[15] RTCA DO-181B *MOPS for ATCRBS/Mode S Airborne Equipment*, RTCA, Inc 1999.

[16] P. Misra, private communication, 24 Jan 2001.

[17] J. Gertz, "Fundamentals of Mode S Parity Coding," MIT Lincoln Laboratory Group 92 Project Report ATC-117, 2 Apr 1984.

[18] R. Wells, *Applied Coding and Information Theory for Engineers*, Prentice Hall, Inc. 1999.

[19] V. Orlando, private communication (member of SICASP for ICAO), 24 Jan 2001.

[20] Z. Ramzan, private communication, 16 Aug 2000.

[21] M. Caloyannides, "Encryption Wars: Early Battles," *IEEE Spectrum,* April 2000.

[22] M. Bellare, and P. Rogaway, "On the Construction of Variable-Input-Length Ciphers," http://www-cse.ucsd.edu/users/mihir/. 22 Apr 1999.

[23] B. Schneier, and D. Whiting, "Fast Software Encryption: Designing Encryption Algorithms for Optimal Software Speed on the Intel Pentium Processor" http://www.counterpane.com/publish-1997.html. Jan 1997.

[24] M. Blaze, W. Diffie, R. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Wiener "Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security," http://theory.lcs.mit.edu/~rivest/publications.html. Jan 1996.

[25] I. Goldberg and D. Wagner, "Architectural Considerations for Cryptanalytic Hardware," http://www.cs.berkeley.edu/~daw/papers/ 1996.

[26] http://kremlinencrypt.com/crypto/algorithms.html

[27] http://www.totalb.com/~mikeg/ce/cryptography.html