Securing Asynchronous Transfer Mode (ATM)

Shaw Chuang

Computer Laboratory

Cambridge University

University of Cambridge
Computer Laboratory

Securing Asynchronous Transfer Mode

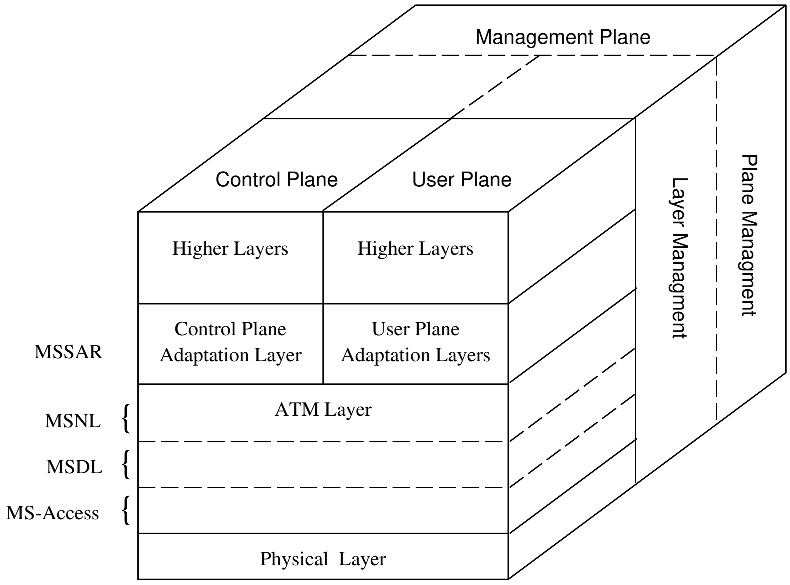
- Introduction to Asynchronous Transfer Mode
- Environment
- Control architecture
- Data plane integrity and confidentiality
- Secure bootstrap
- Security issues with signaling
- Other issues

University of Cambridge
Computer Laboratory

Introduction to Asynchronous Transfer Mode

The Asynchronous Transfer Mode is the transfer mode for implementing BISDN

BISDN Protocol Reference Model



University of Cambridge
Computer Laboratory

Introduction to ATM: Data Plane

- Virtual Circuit (VC) based
- Provides for the transfer of user information
  - Flow control
  - Recovery from error
- Information to be transferred is packed into fixed-size slots called cells

University of Cambridge
Computer Laboratory

Introduction to ATM: Data Plane cont....

UNI Cell format

BIT

8 7 6 5	4 3 2	1	Octet
GFC	VPI		1
VPI	VCI		2
VCI			3
VCI	PT	CLP	4
HEC			5

- 5 octet header
  Carries information that pertains to the ATM layer functionality itself
- Fixed size
- Easy to use
- 48 octet information field

University of Cambridge
Computer Laboratory

Introduction to ATM: Control Plane

- Responsible for the call control and connection control function
- These are all signaling functions which are necessary to
  - setup
  - supervise
  - release

  a call or connection
- Physical layer and the ATM layer are the same for the control and user plane.

University of Cambridge
Computer Laboratory

Introduction to ATM: Management Plane

2 types of functions:

- Layer Management function
  - Perform the management functions relating to resources and parameters residing in its protocol entity (e.g. Meta-signaling)
  - Handles the specific OAM information flows for each layer
- Plane Management function
  - All the management function that relate to the whole system
  - Provide coordination between all planes

University of Cambridge
Computer Laboratory

Introduction to ATM: ATM Adaptation Layer

Enhanced adaptation of service provided by the ATM layer to the requirements of the higher layer
2 sublayers:

- Segmentation and Reassembly (SAR)

  Segmentation at the transmitter side and reassembly at the receiver side

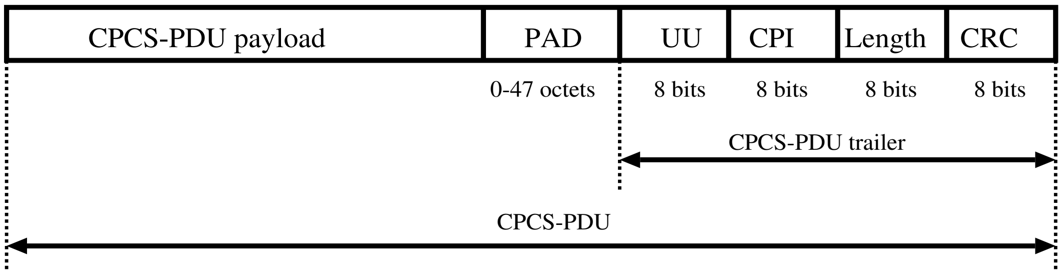
- Convergence sublayer (CS)

  Service dependent and provide the AAL service at the AAL SAP

e.g. AAL1, AAL2, AAL3/4, AAL5

Introduction to ATM: AAL5

- Applied to variable bit rate sources without a timing relation between source and destination
- reduced overhead
- Does not support a multiplexing function
- use for signaling and frame relay over ATM
- For data applications
- Very widely used

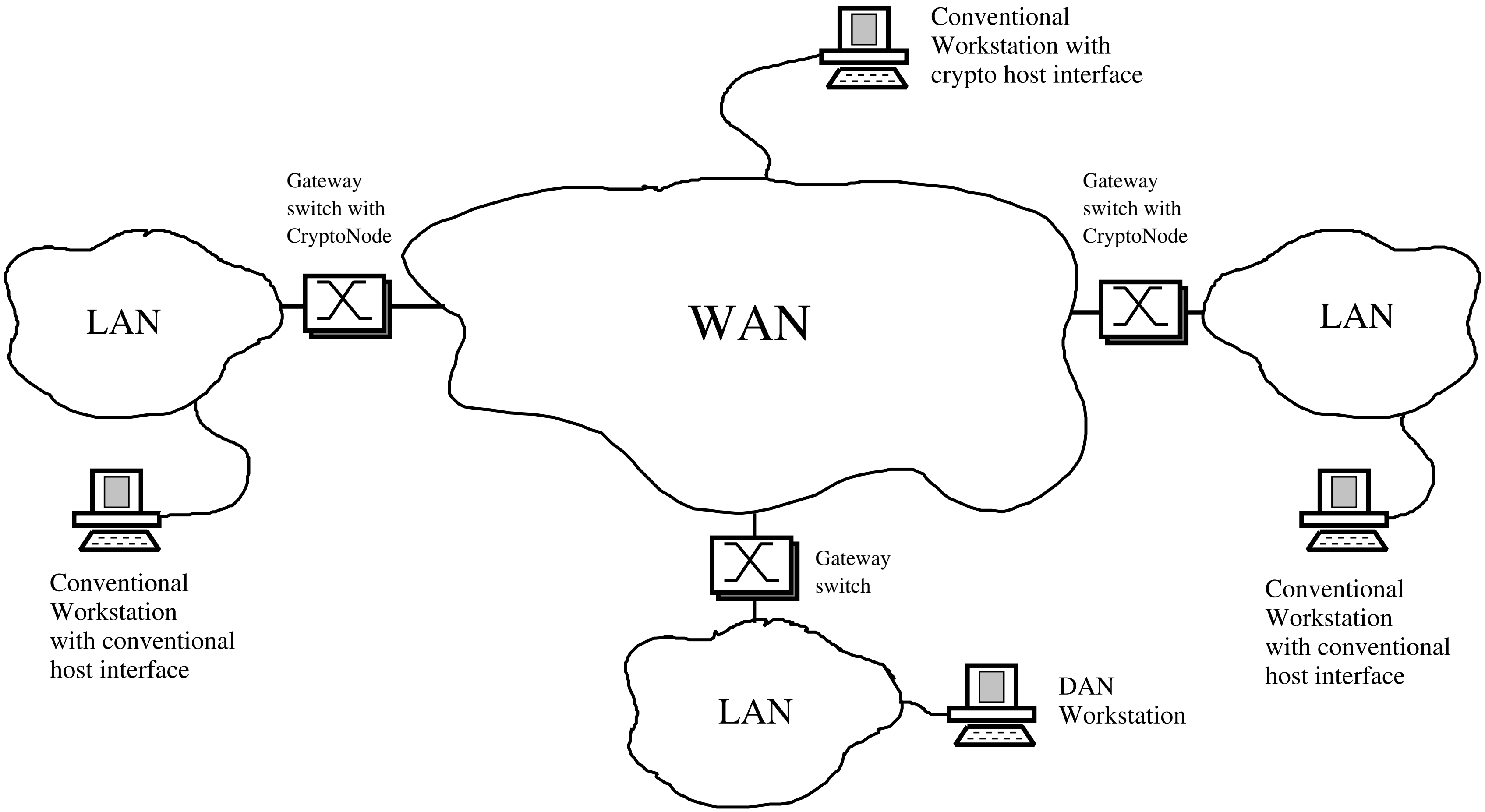


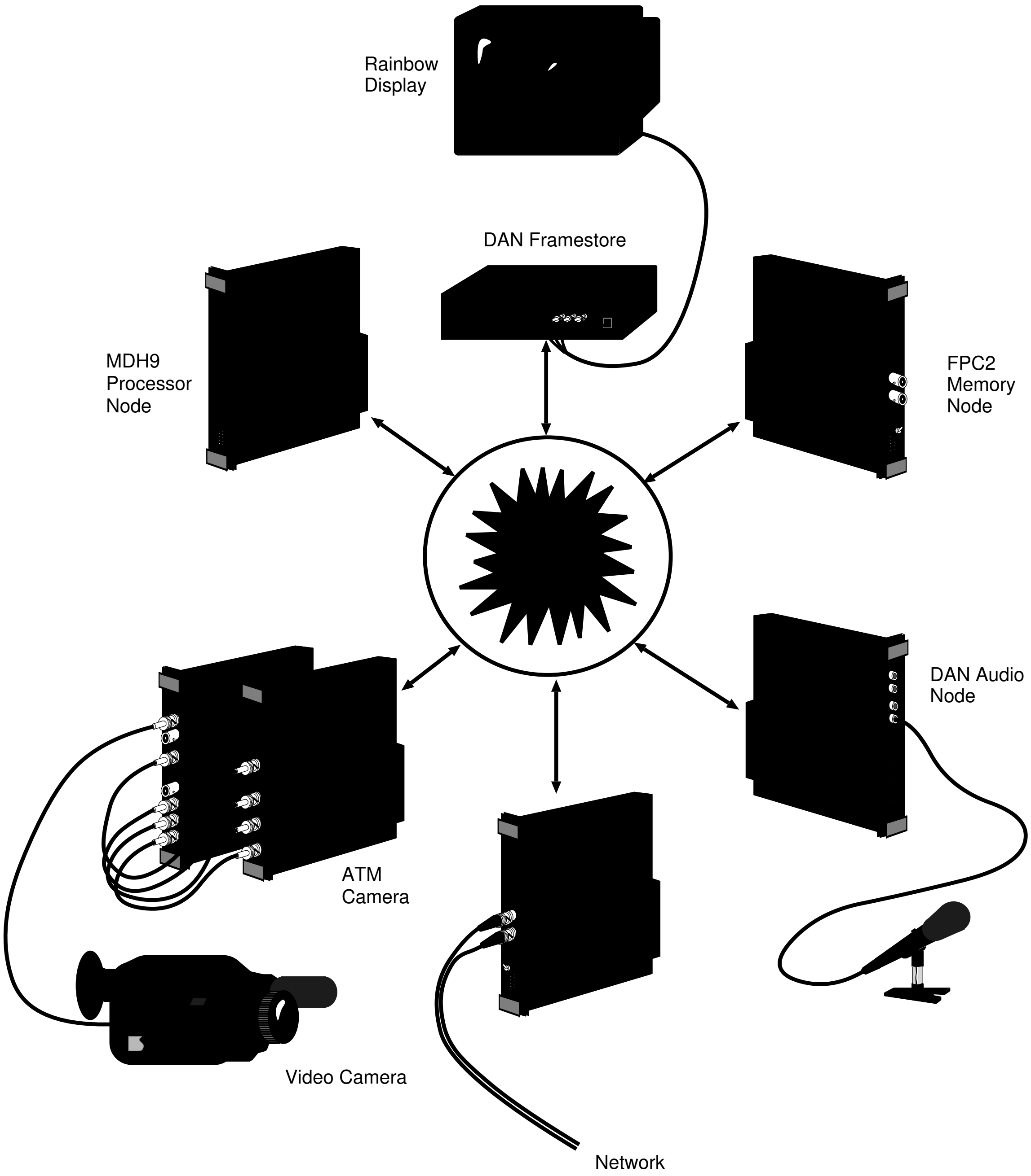
University of Cambridge
Computer Laboratory

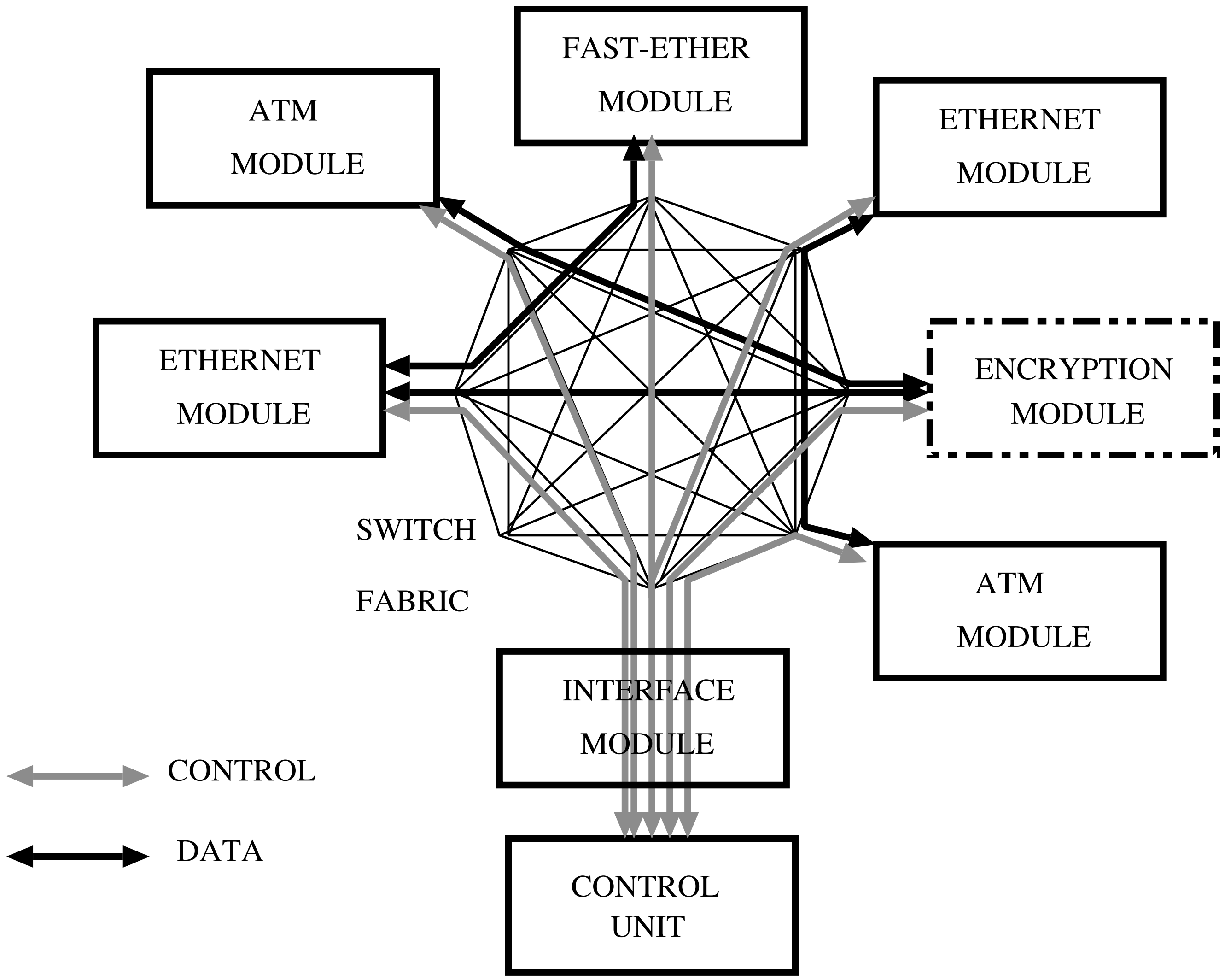
Network Environment

- Workstations with conventional host interface
  Crypto software in device driver or middleware
- Workstations with secure host interface
- Multimedia workstations e.g. Desk Area Network
- ATM switches with crypto device

University of Cambridge
Computer Laboratory


Conventional
Workstation with
crypto host interface
Gateway
switch with
CryptoNode
Gateway
switch with
CryptoNode
LAN
WAN
LAN
Conventional
Workstation
with conventional
host interface
Gateway
switch
Conventional
Workstation
with conventional
host interface
LAN
DAN
Workstation


Rainbow
Display
DAN Framestore
MDH9
Processor
Node
FPC2
Memory
Node
DAN Audio
Node
ATM
Camera
Video Camera
Network

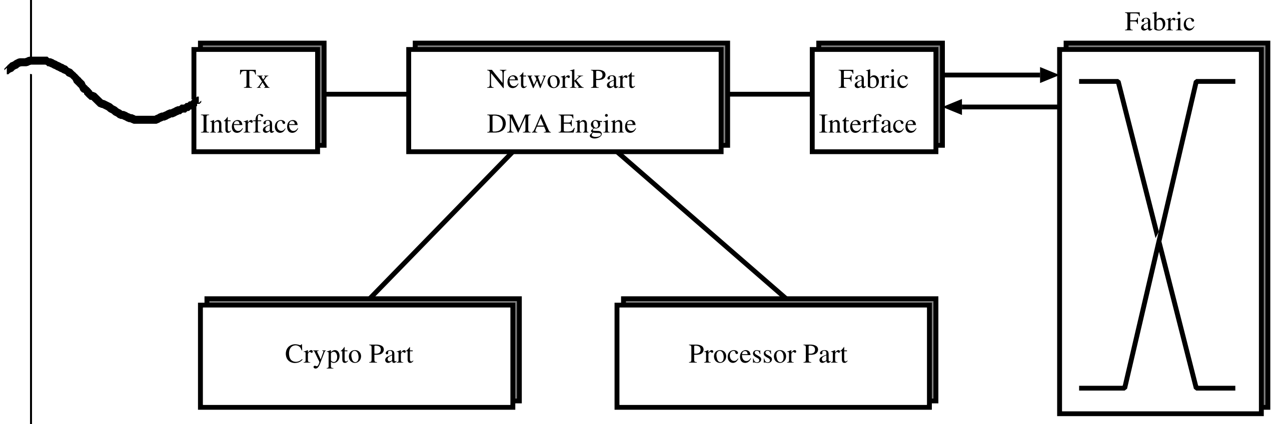

FAST-ETHER
MODULE
ATM
MODULE
ETHERNET
MODULE
ETHERNET
MODULE
ENCRYPTION
MODULE
SWITCH
FABRIC
ATM
MODULE
INTERFACE
MODULE
CONTROL
DATA
CONTROL
UNIT

CryptoNode

- Plug straight into the switch fabric
- Can act as DAN secure co-processor
- Can also be used in DAN-based routers
- Logically consists of three parts
  - Network
  - Processor
  - Crypto
- Simulated using the Fairisle Port Controller

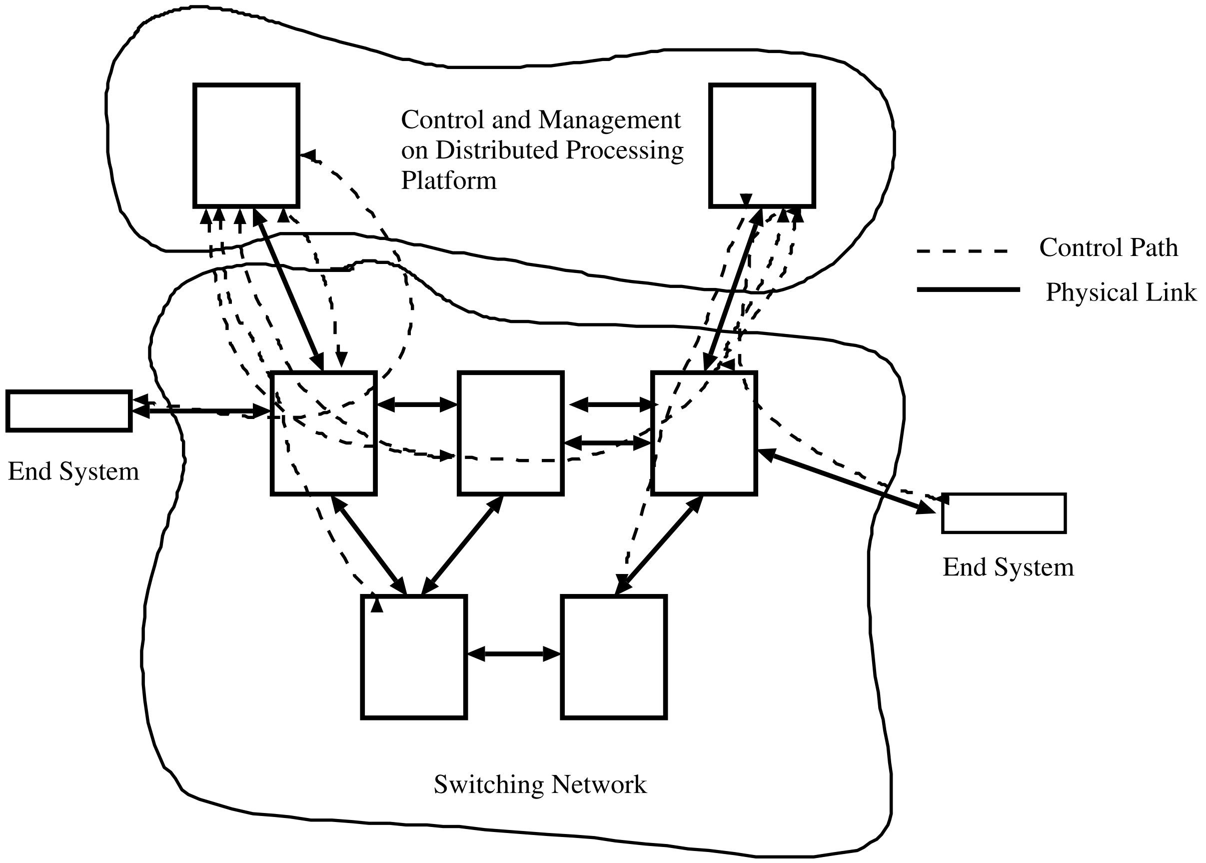
University of Cambridge
Computer Laboratory

CryptoNode: Block Diagram



University of Cambridge
Computer Laboratory

Control architecture



University of Cambridge
Computer Laboratory

Data Plane

Threats to the Data Plane

- Confidentiality
- Integrity

Problems with Securing Data Plane

- Granularity of protection
- Synchronisation points
- Integrity check value/point
- Key lifetime

University of Cambridge
Computer Laboratory

Data Plane: Granularity of protection

Granularity of protection

- Cell multiplexing
- per VCI protection
- Context loading rate to crypto devices

University of Cambridge
Computer Laboratory

Data Plane: Synchronisation Point

WHERE ?

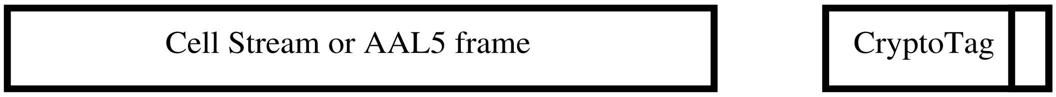
- Periodic: Cell stream
- Encapsulation
- PDU boundaries: AAL5 frame
  - Make better sense if we do integrity check
  - AAL5 is most widely used AAL

WHAT FORM ?

- Need in-band mechanism
  - OAM cell
  - Cell with special payload pattern
  - Special AAL frame
    Set UU bits

University of Cambridge
Computer Laboratory

Data Plane: Synchronisation Information



HOW BIG ?

Single cell AAL5 frame: CryptoTag Cell

- Easy Detection
- Network efficiency
- minimum buffering
- Compatible with all systems

University of Cambridge
Computer Laboratory

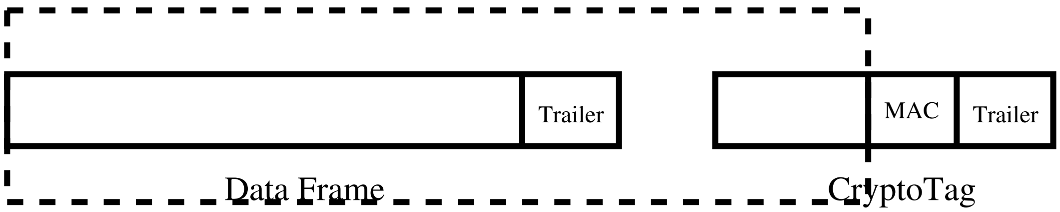
Data Plane: Synchronisation Information cont....

WHAT IS IN IT ?

We have 40 bytes to play with

Key Hint	64 bits
Confidentiality IV	64 bits
Integrity IV	64 bits
MAC	128bits

MAC Coverage ?



University of Cambridge
Computer Laboratory

Data Plane: Synchronisation Information cont....

KEY HINT

- Indicate key change
- Can be a structured field

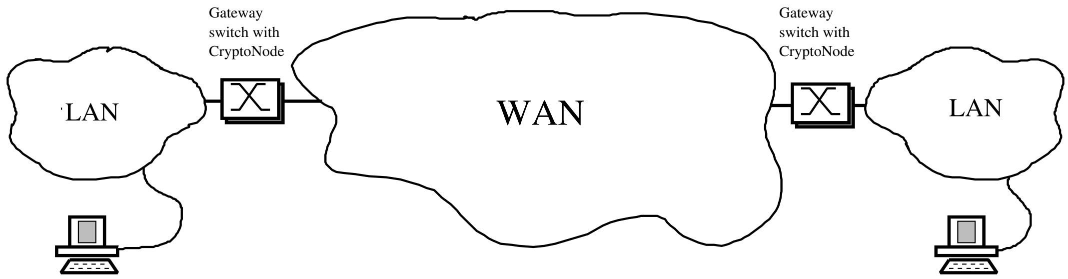
Confidentiality Master Key	Confidentiality Session Key	Integrity Master Key	Integrity Session Key

- Generate interrupt when change
- Interpreted by higher level software
  - Index into a table of keys
  - Encrypt (K_m, K) where K_m is the master key
- Per VCI queuing

University of Cambridge
Computer Laboratory

Data Plane: CRC

AAL5 needs re-calculation after ciphering payload



CRC' = CRC (*Plaintext* XOR *Ciphertext*)
New CRC = (*OLD CRC*) XOR (*CRC'*)

University of Cambridge
Computer Laboratory

Data Plane: End System

- Crypto Host Interface
  Cut-through processing
- Secure Device Driver
- Application Middleware
  - Need to deliver UU bits to application
  - Expose session key to application

University of Cambridge
Computer Laboratory

Secure Booting

Threats

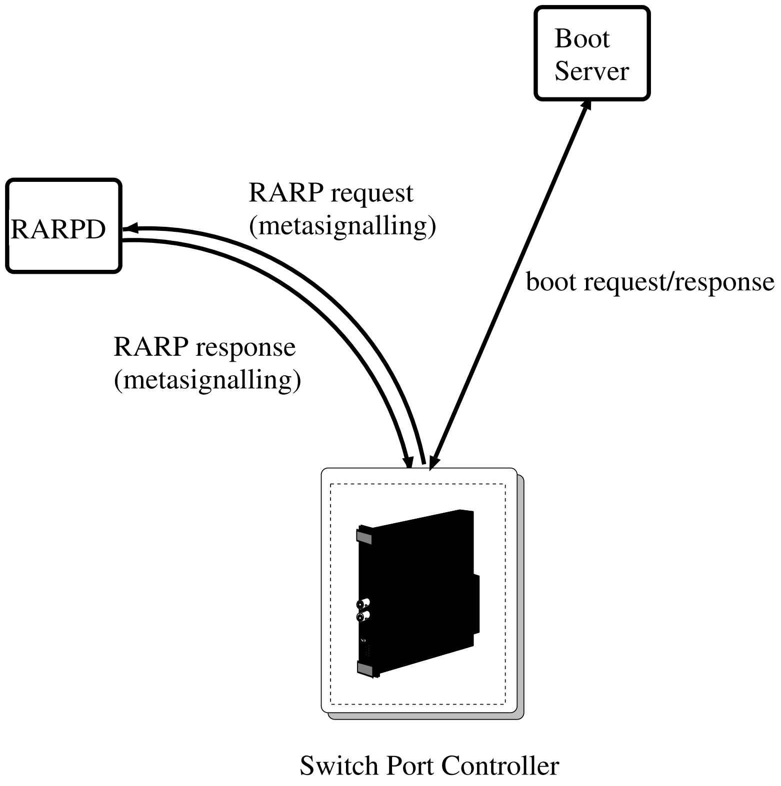
- RARP information modification
- Boot server information modification
- Boot server identity modification
- Boot code modification

Protection

- RARP information integrity and authenticity
- Boot server authenticity
- Boot code integrity and authenticity

University of Cambridge
Computer Laboratory

Secure Booting Scenario



University of Cambridge
Computer Laboratory

Secure Booting cont...

- Machine kept secret, e.g. DES key
- Minimum use of this secure to prevent cryptanalysis
- Use symmetric crypto algorithm due to size limitation of metasignaling cell

Trust Model

- RARPD trusted; RARPD shared the secret
- Owner of the switch create the code to run; i.e. he knows the secret

University of Cambridge
Computer Laboratory

Secure Booting cont...

Phase zero

- Owner sign two public key certificates using the DES key
  1. Owner public key
  2. Boot server public key
- Owner encrypt a new public key pair using the DES key
- Owner sign the boot code with his private key

Phase one

- RARPD request and response have MAC with the DES key
- RARPD request with challenge to prevent replay
- If MAC fail, machine repeat rarping
- If successful, we have got own address and boot server address

University of Cambridge
Computer Laboratory

Secure Booting cont...

Phase two

- Connect to 'boot server'
- Send a challenge with the boot request
- Boot server sign the response and the boot code with his public key
- Also send the two public key certificate, and the encrypted public key pair

$$S \rightarrow B : N_s$$

$$B \rightarrow S : \{N_s, S, (code)K_M^{-1}, \{K_{ST}, K_S, K_S^{-1}\}_{K_{RS}}\}_{K_B^{-1}}, \{S, K_A, K_B\}_{K_{RS}}$$

$$\{K_{ST}, K_S, K_S^{-1}\}_{K_{RS}} : Both\ integrity\ and\ confidentiality\ protected$$

University of Cambridge
Computer Laboratory

Secure Booting cont...

Difficult to generate good random number

- Have a counter, C, in non-volatile memory
- use K_{ST} (a symmetric key) to generate the random number
  i.e. N_S = K_{ST}(C)

Need to protect K_{RS} from leakage by secondary boot code either accidental or malicious

- Read-once mapping for K_{RS}
- Primary boot code erase, K_{RS}, from memory before starting secondary code

University of Cambridge
Computer Laboratory

Metasignaling

Other metasignaling functions:

- initial control connection setup
- connection disconnect

Authentication should be done by the entities explicitly.

University of Cambridge
Computer Laboratory

Security issues with signaling

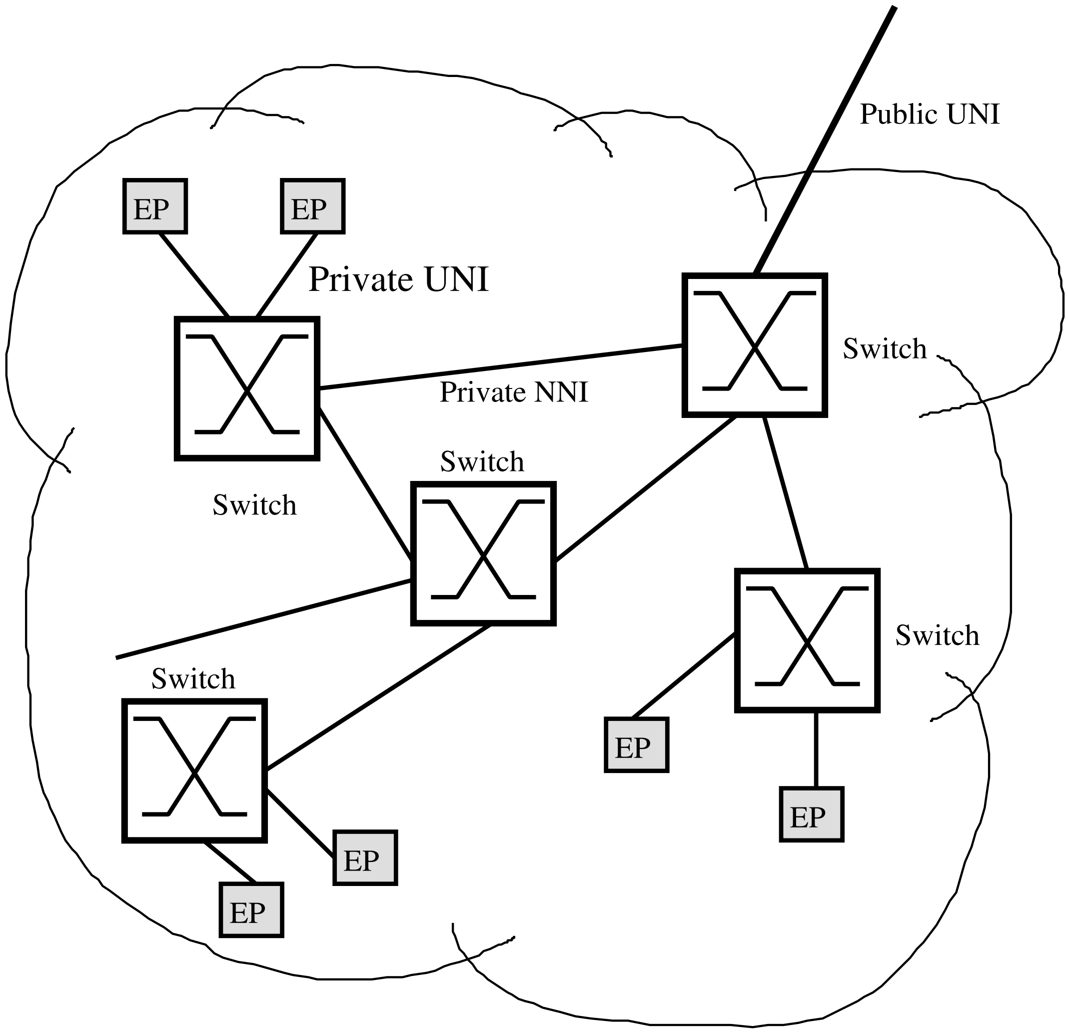
- Authenticated signaling
- Secure Signalling
- Signalling Security

University of Cambridge
Computer Laboratory

Authenticated Signalling and Key Exchange

1. end user to end user,
2. user to network,
3. third party user to remote network, and
4. network to network.
5. switch to control manager

University of Cambridge
Computer Laboratory



University of Cambridge
Computer Laboratory

Key Exchange constraints

- avoid timestamp
- preferable 2-way (UNI3), max 3-ways (MSNL)
- Speed:
  - constraint by call setup time
  - especially bad when WAN
- Cross-domain call wants non-repudiation
  Public key ?
- uni-directional key
- May exchange a set of keys for efficiency

University of Cambridge
Computer Laboratory

Protocol 1

$$A \rightarrow B : E_B(N_A)$$

$$B \rightarrow A : E_A(N_B), S_B(E_A(B, N_A, K_{BA}, C)$$

$$A \rightarrow B : S_A(E_B(A, N_B, K_{AB}, C1)$$

University of Cambridge
Computer Laboratory

Protocol 2

$$A \rightarrow B : \alpha,\ p,\ \alpha^x\ mod\ p$$

$$B \rightarrow A : \alpha^y, CERT_B, E_K(S_B\{\alpha^y,\ \alpha^x\})$$

$$A \rightarrow B : E_K(S_A\{\ \alpha^x, \alpha^y\})$$

$$where\ CERT_A\ =\ (Alice,\ P_A,\ \alpha,\ p,\ S_T\{Alice,\ P_A,\ \alpha,\ p\})$$

University of Cambridge
Computer Laboratory

Protocol 3

X.509

- 1-3 phases
- 1/2 phase version needs synchronised clock

$$A \to B : \{\ A,\ R_A,\ T_A,\ \{K_{AB},\ A\}_{K_B}\ \}_{K_A-1}$$

$$B \to A : \{\ B,\ R_B,\ T_B,\ \{K_{BA},\ B\}_{K_A}\ \}_{K_B-1}$$

$$A \to B : \{R_B,\ B\}S_A$$

University of Cambridge
Computer Laboratory

Robust Call Setup

- Handle Byzantine failure
- Denial of service
- Restraint broadcast

University of Cambridge
Computer Laboratory

Other issues

- Denial of service countermeasure
- Management Plane
- Secure tariffing mechanism

University of Cambridge
Computer Laboratory