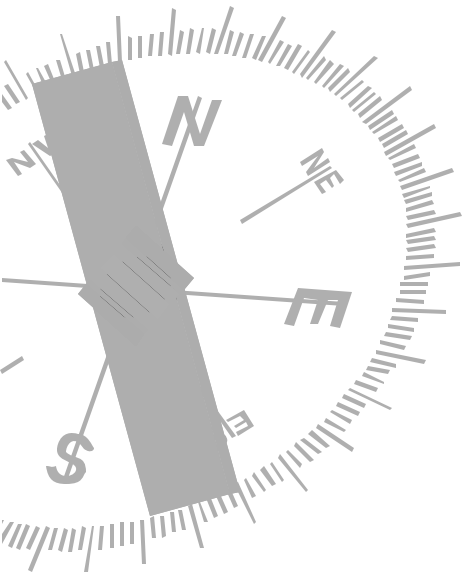


# Secure Broadcasting

Camelia Al-Najjar  
Bryan Graham



# Sources

? *The BiBa One-Time Signature and Broadcast Authentication Protocol* by Adrian Perrig

? *Key Management for Encrypted Broadcast*  
by Avishai Wool



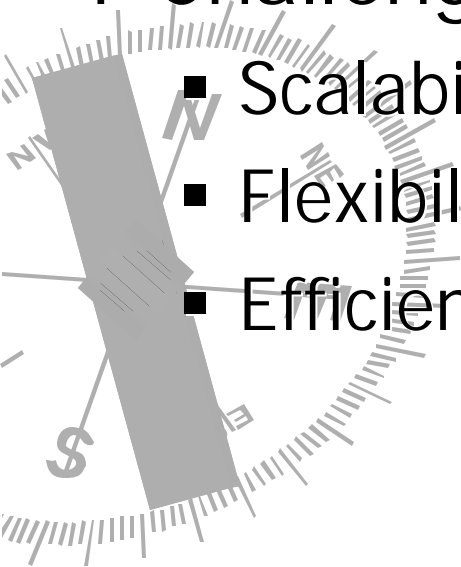
# Secure Broadcasting

## ? Requirements

- Authentication
- Privacy

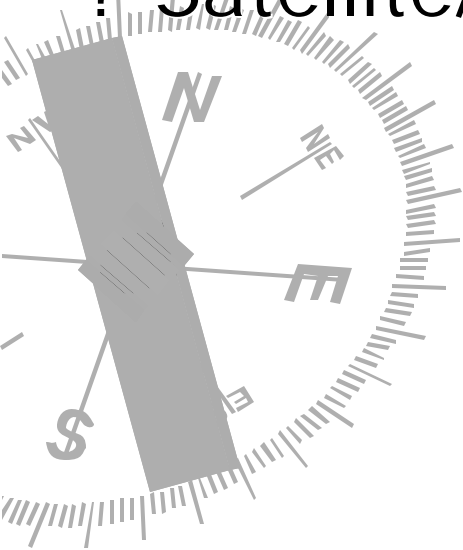
## ? Challenges

- Scalability
- Flexibility
- Efficiency



# Secure Broadcasting Uses

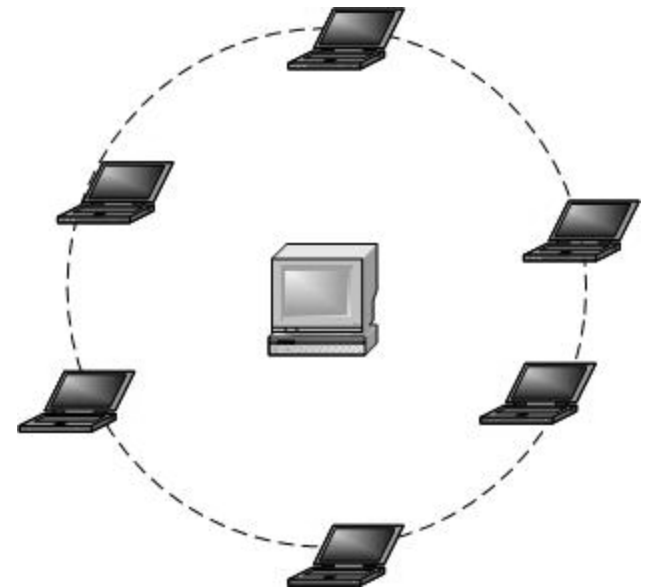
- ? Wireless Domain
- ? Audio/Video Conferencing
- ? News/Stock Tickers
- ? Satellite/Cable TV Delivery



# Wireless Domain

- ? Broadcast by virtue of being wireless
- ? Usually one/few "intended" receivers
- ? Often similar to wired case
- ? Problems

- Dynamic Nature
- Limited Resources
- Higher error rate



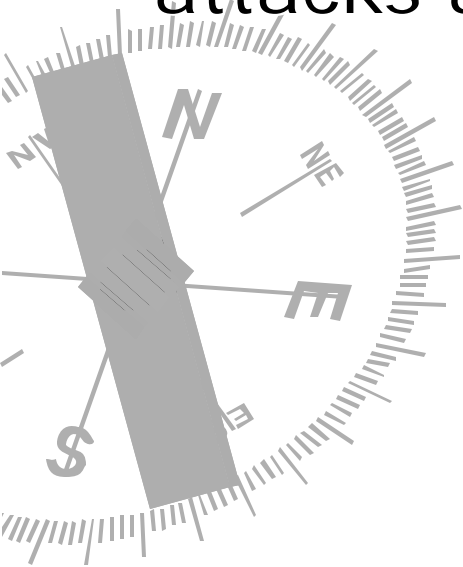
# Audio/Video Conferencing

- ? Usually “broadcast” for convenience
- ? Only certain “intended” recipients
- ? Typically more of multicast case
- ? Efficiency required for quality



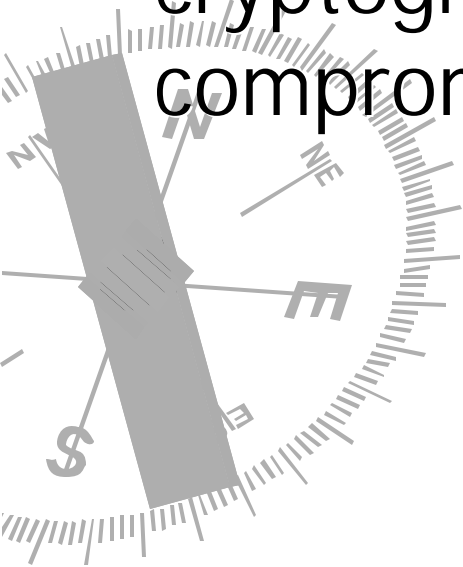
# News/Stock Tickers

- ? Data usually not encrypted
- ? Authentication is key
- ? “Man-in-the-middle” and impersonation attacks are most serious problems



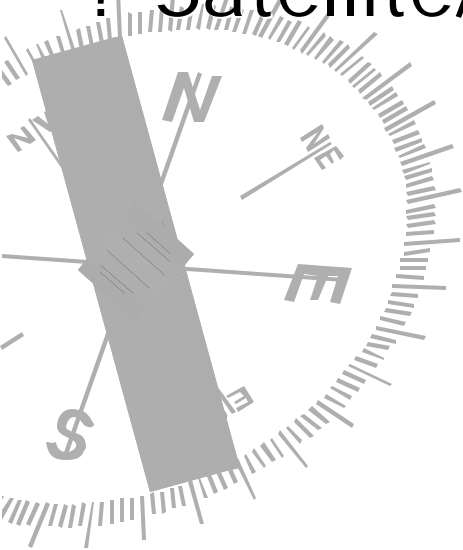
# Satellite/Cable TV Delivery

- ? All channels are broadcast
- ? Subscribed Customers – Subscribed Data
- ? Similar adversaries to standard cryptography (known plain-text, compromised node)



# Secure Broadcasting Uses

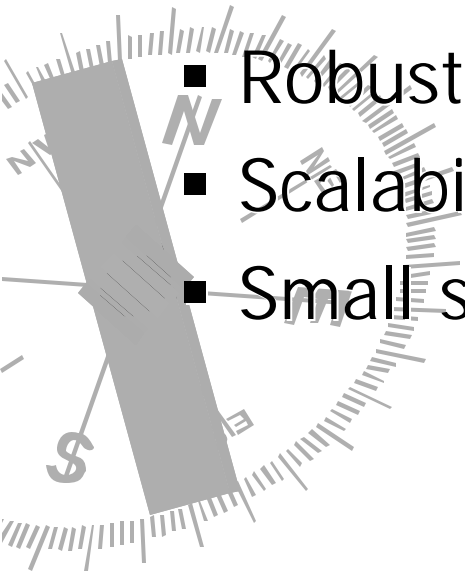
- ? Wireless Domain
- ? Audio/Video Conferencing
- ? News/Stock Tickers
- ? Satellite/Cable TV Delivery



# BiBa Broadcast Authentication

## ? Challenges

- Efficient generation and verification
- Real-time/instant authentication
- Individual message authentication
- Robustness to packet loss
- Scalability
- Small size of authentication information



# BiBa – Bins and Balls

? Sender generates list of SEALS

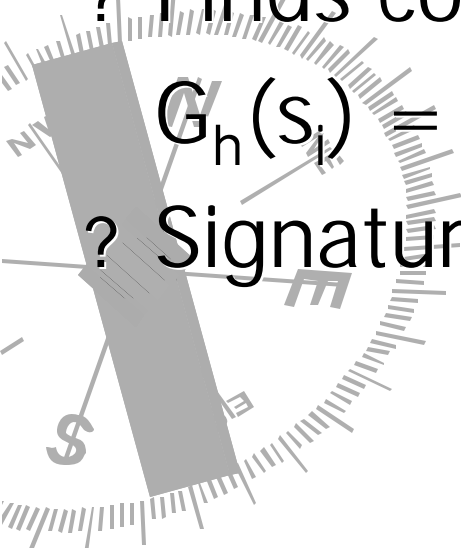
?  $h = H(m)$

?  $G_h(s_1) \dots G_h(s_t)$

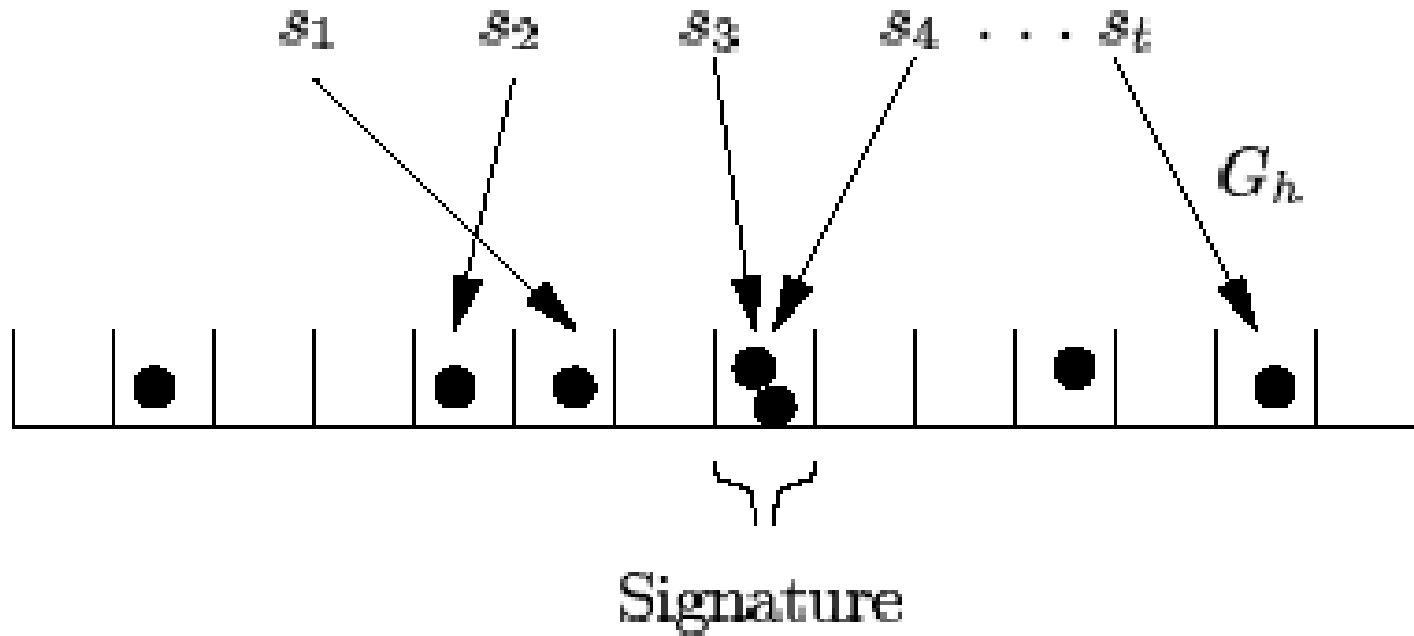
? Finds collision where

$$G_h(s_i) = G_h(s_j) \text{ and } s_i \neq s_j$$

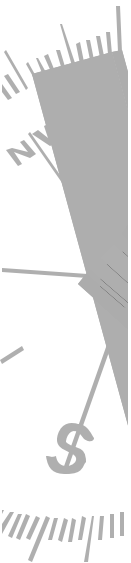
? Signature is  $(s_i, s_j)$



# Bins and Balls

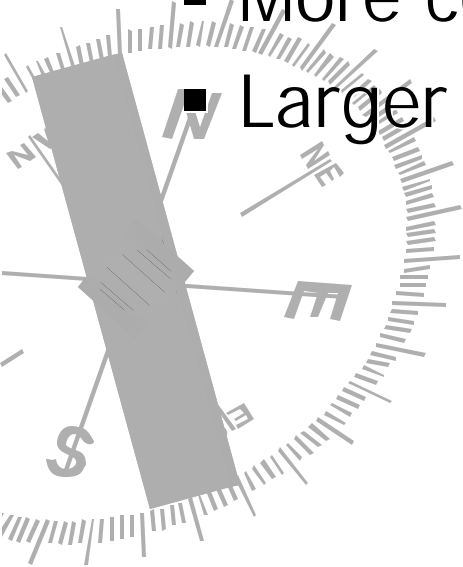


**Figure 1: Basic BiBa scheme**

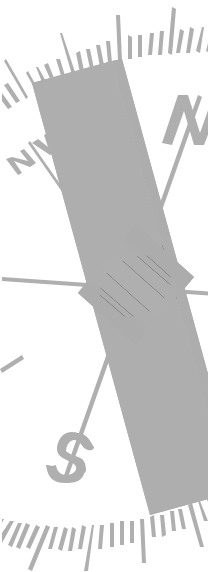
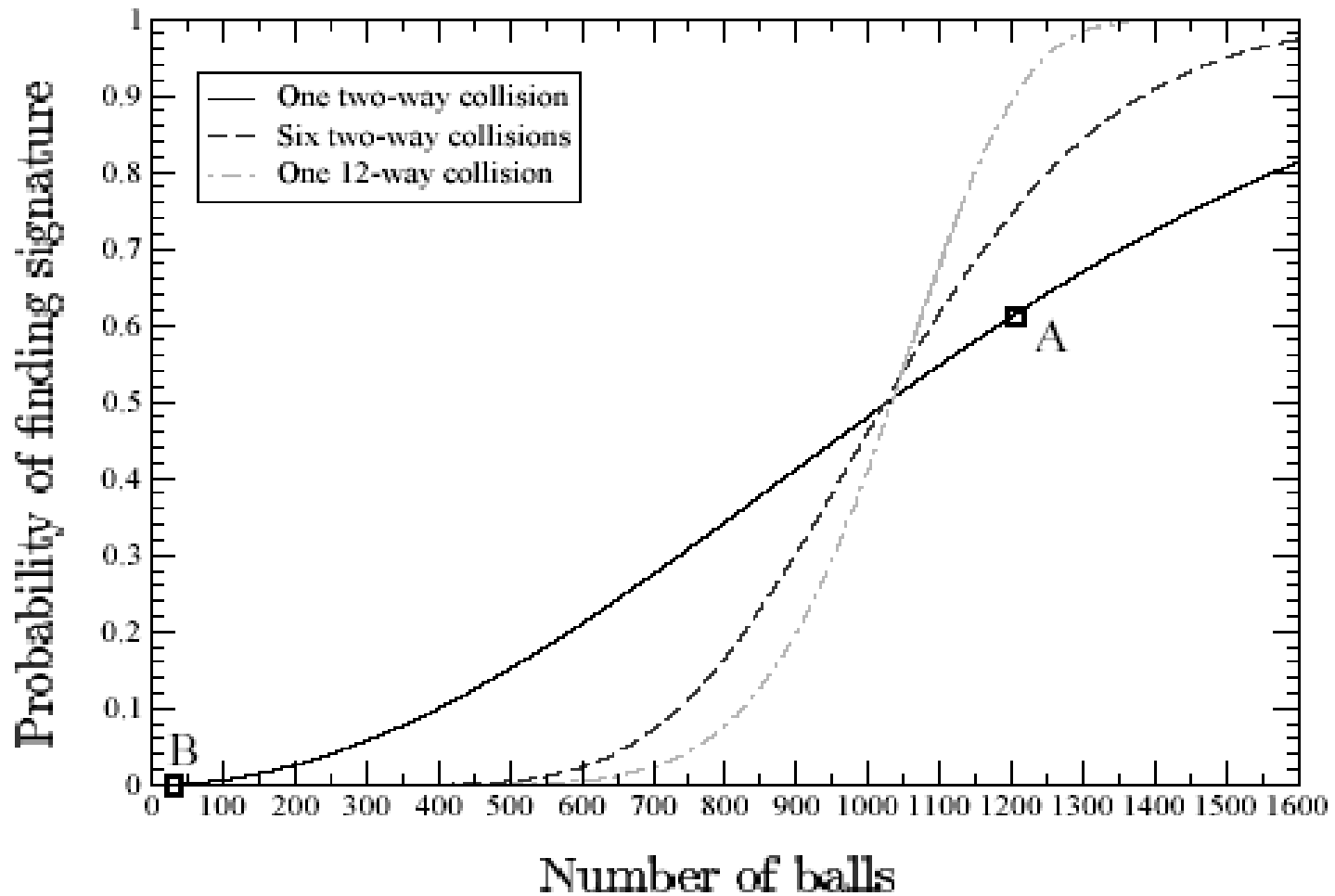


# BiBa Security

- ? Sender knows lots of SEALS
- ? Adversary knows only the disclosed SEALS
- ? Increase Security
  - More collisions
  - Larger collisions

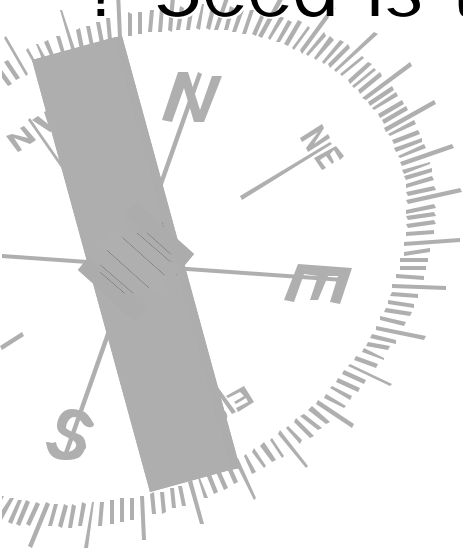


# BiBa Security



# SEALs

- ? Limited number of active disclosed SEALs
- ? SEALs generated using PRF
- ? One-way SEAL chains
- ? Seed is time-based



# SEAL Generation

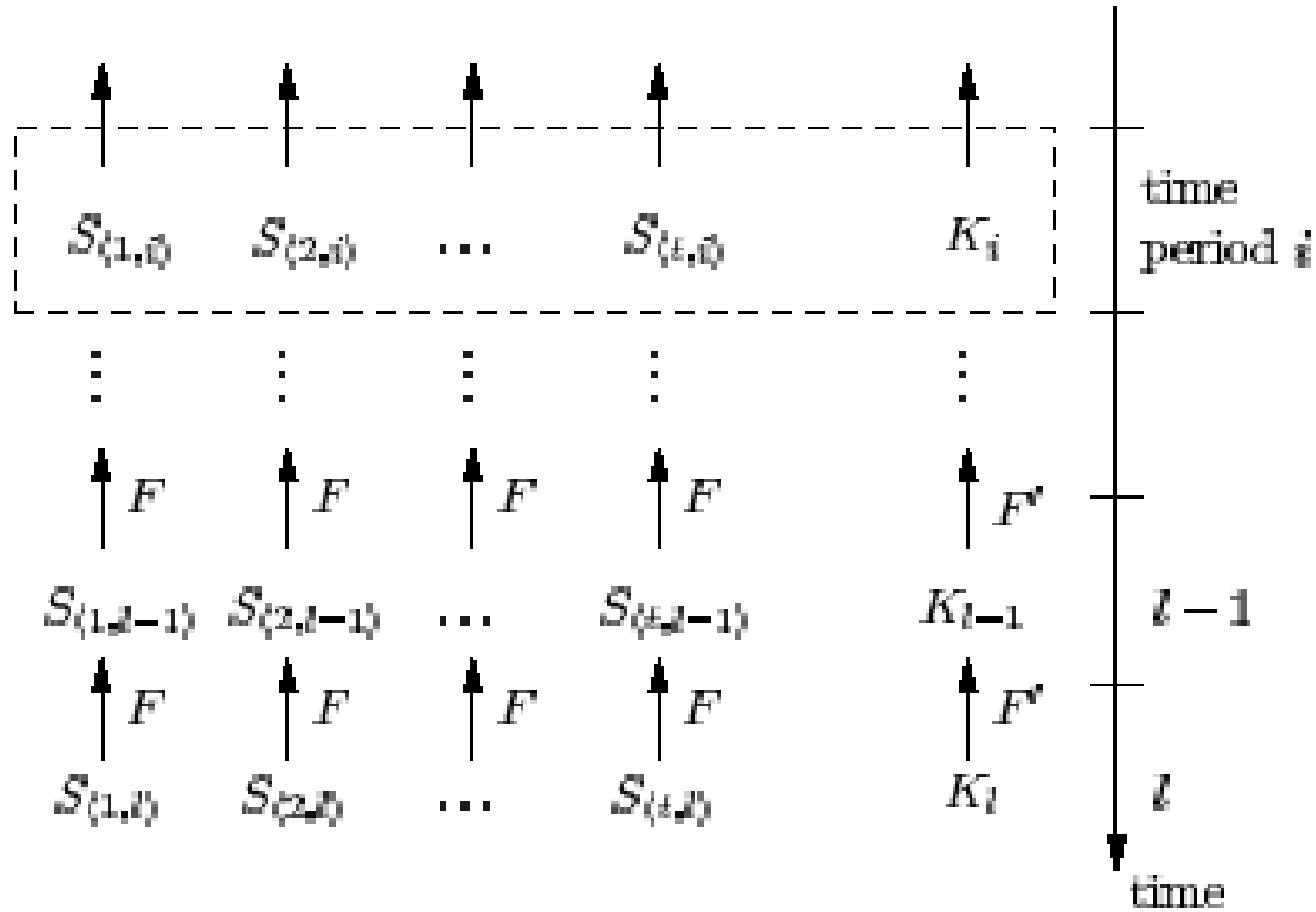
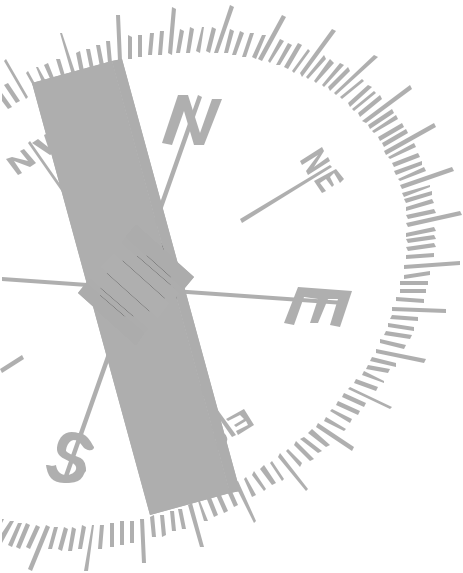


Figure 3: Using one-way chains to construct SEAL

# BiBa Performance

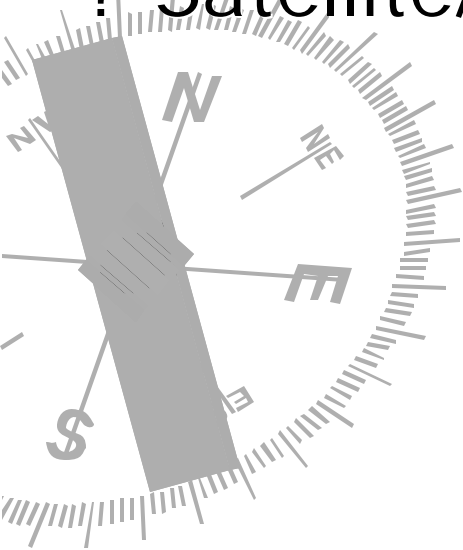
? vs. OpenSSL 1024-bit RSA

- Generation of BiBa signature is 5x faster
- Verification of BiBa signature is 20x faster
- BiBa is easily parallelized



# Secure Broadcasting Uses

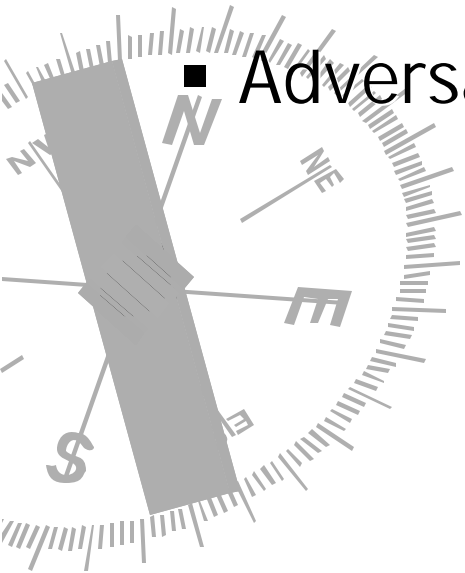
- ? Wireless Domain
- ? Audio/Video Conferencing
- ? News/Stock Tickers
- ? Satellite/Cable TV Delivery



# TV Broadcasting

## ? Challenges

- Flexibility
- Security
- Hardware Limitations – RAM and CPU
- Adversarial Considerations



# Terminology

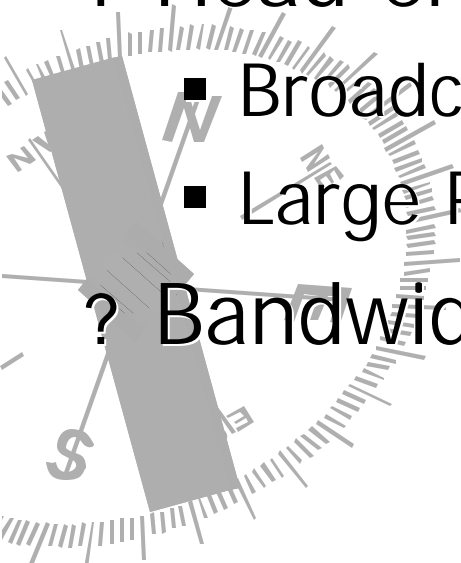
? STT – set-top terminals, smart-chip

- Very limited secure RAM
- Possibly limited CPU power

? Head-end

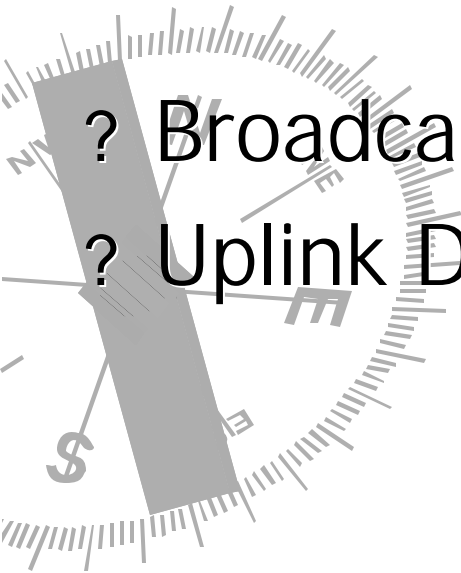
- Broadcasts the data
- Large Processing Power

? Bandwidth is precious



# Key Management

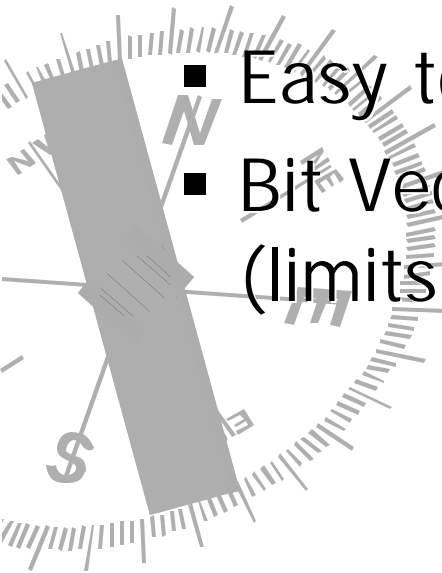
- ? Establishment Key
- ? Periodic Key
- ? Individual Program Key
- ? Broadcast Distribution
- ? Uplink Distribution



# Current Technology

## ? Bit Vector

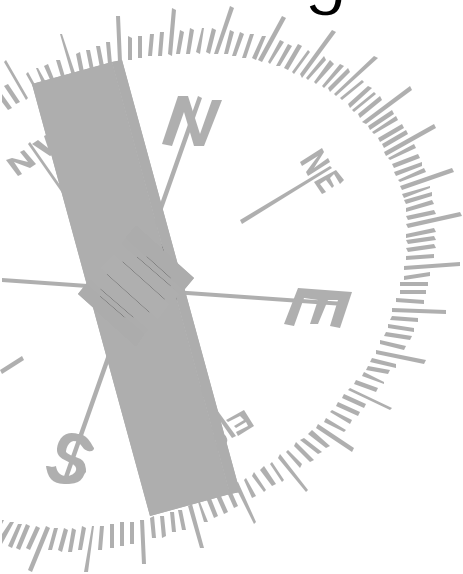
- All programs use same key
- Bit Vector determines what can be seen
- Good Flexibility
- Easy to break
- Bit Vector must be stored in secure memory (limits scalability)



# Current Technology

## ? Block by Block

- $n$  disjoint blocks each with different key
- Better security, worse flexibility
- Higher overhead



# ExtHeader Scheme

- ? Header for each package
- ?  $E_s$  algorithm can be stronger than  $Enc_K$
- ? Overhead problem
- ? Bandwidth vs. Allowable Delay

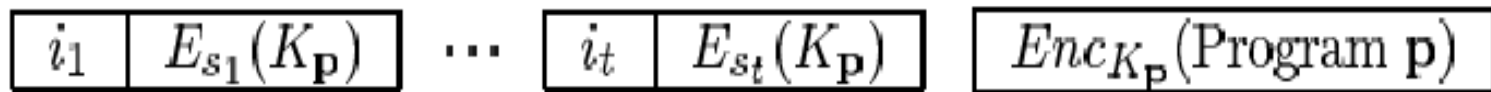
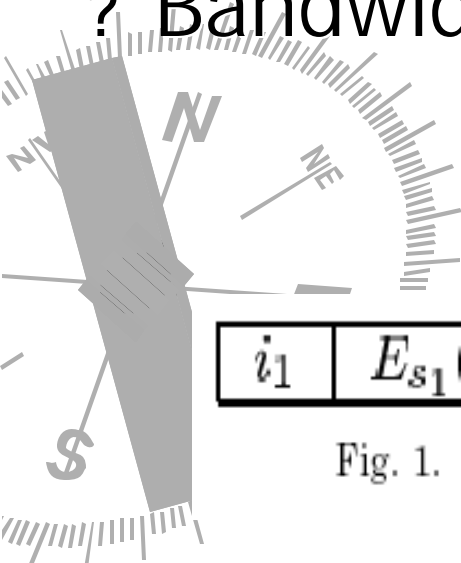
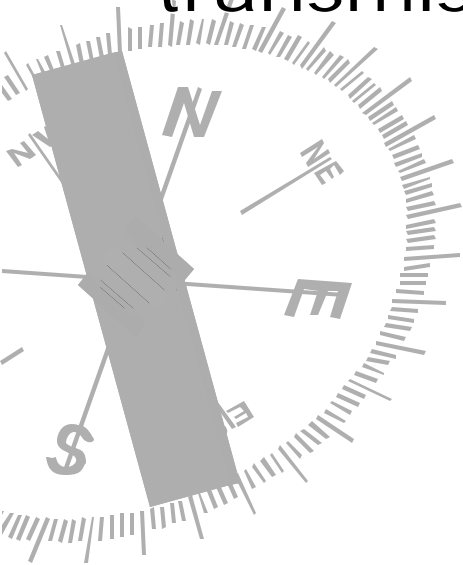


Fig. 1. The headers of a program  $p$  which belongs to packages  $i_1, \dots, i_t$ .

# Improving ExtHeader

- ? Additional STT memory
- ? Cache encrypted keys
- ? Supplemental channel for header transmission



# Vspace Scheme

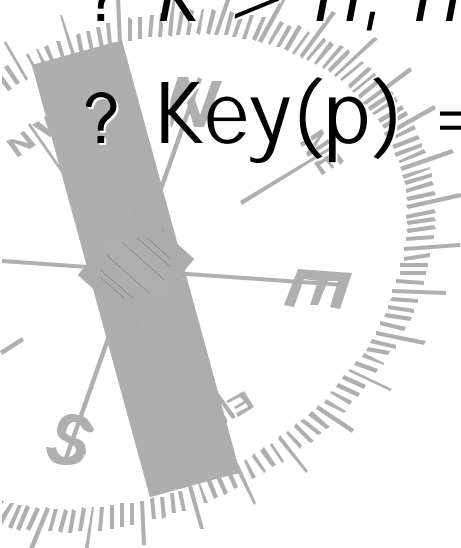
?  $n$  bit CID  $p$

?  $k$  bit encryption key

? Sender has  $k \times n$  matrix  $M$

?  $k > n$ ,  $n$  linearly independent columns

?  $\text{Key}(p) = M^*p$



# Linear Subspace Paradigm

- ? Alice has  $\text{Key}(p_1)$  and  $\text{Key}(p_2)$
- ?  $\text{Key}(p_1) \oplus \text{Key}(p_2) = \text{Key}(p_1 \oplus p_2)$
- ? Program sets are linear subspaces
- ? Flexibility – packages are of size  $2^n - 1$
- ?  $\text{CID } p = 0$  for network channels



# Receiver Side

?  $r$  dimensional subspace of  $M$  is  $U$

?  $U$  has basis  $B$

$$Bx = p, \quad \text{for some } x$$

? Can be solved with  $B^{-1}$

?  $K$  contains the key data

$$K = MB$$



# Receiver Side

$$Dec = Kx = MBx = Mp = Key(p)$$

$$\text{Since } x = B^{-1}p$$

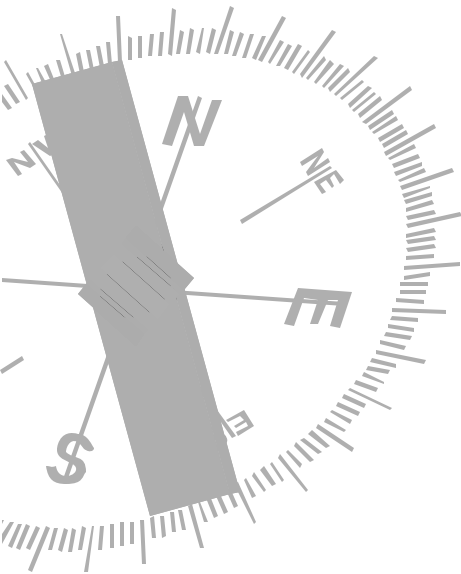
$$D = KB^{-1}$$

$$\text{Then } Dec = D^*p$$



# Creating CIDs and Packages

? Packages use Hierarchical structure



[2 bits]

**0:** Bonus ...

**1:** Movies ...

**2:** Sports [5 bits]

**2.0:** Baseball ...

**2.1:** Football ...

**2.2:** Basketball [2 bits]

**2.2.0:** College

**2.2.1:** Professional

**2.2.2:** European

**2.2.3:**

...

**2.31:**

**3:** Other ...

# Tree Scheme

- ? Similar to Huffman Codes
- ? Binary Tree of Hash values

$$\text{Key}(p) = H_{p_1}(H_{p_2}(\dots H_{p_n}(m)\dots))$$

- ? Program sets are given using sub-trees



# References

- ? Perrig, Adrian. *The BiBa One-Time Signature and Broadcast Authentication Protocol.*
- ? Wool, Avishai. *Key Management for Encrypted Broadcast.*
- ? Liu, Donggang, et al. *Efficient Self-Healing Group Key Distribution with Revocation Capability.*
- ? Blundo, C., et al. *Generalized Beimel-Chor Schemes for Broadcast Encryption and Interactive Key Distribution.*
- ? Blundo, C. et al. *Multiple Key Distribution Maintaining User Anonymity via Broadcast Channels.*
- ? Gong, Li. *New Protocols for Third-Party-Based Authentication and Secure Broadcast.*



# Appendix: Evaluation

## ? Tamper-Proof

- ExtHeader – several copies of the same key
- Vspace – could exploit linearity
- Tree – known plaintext on video stream

## ? Non-Tamper-Proof

- ExtHeader – Covering algo resistant to known plaintext
- Vspace – multiple STTs could allow for determining the parent subspace
- Tree – Inefficient to merge multiple entitlements

