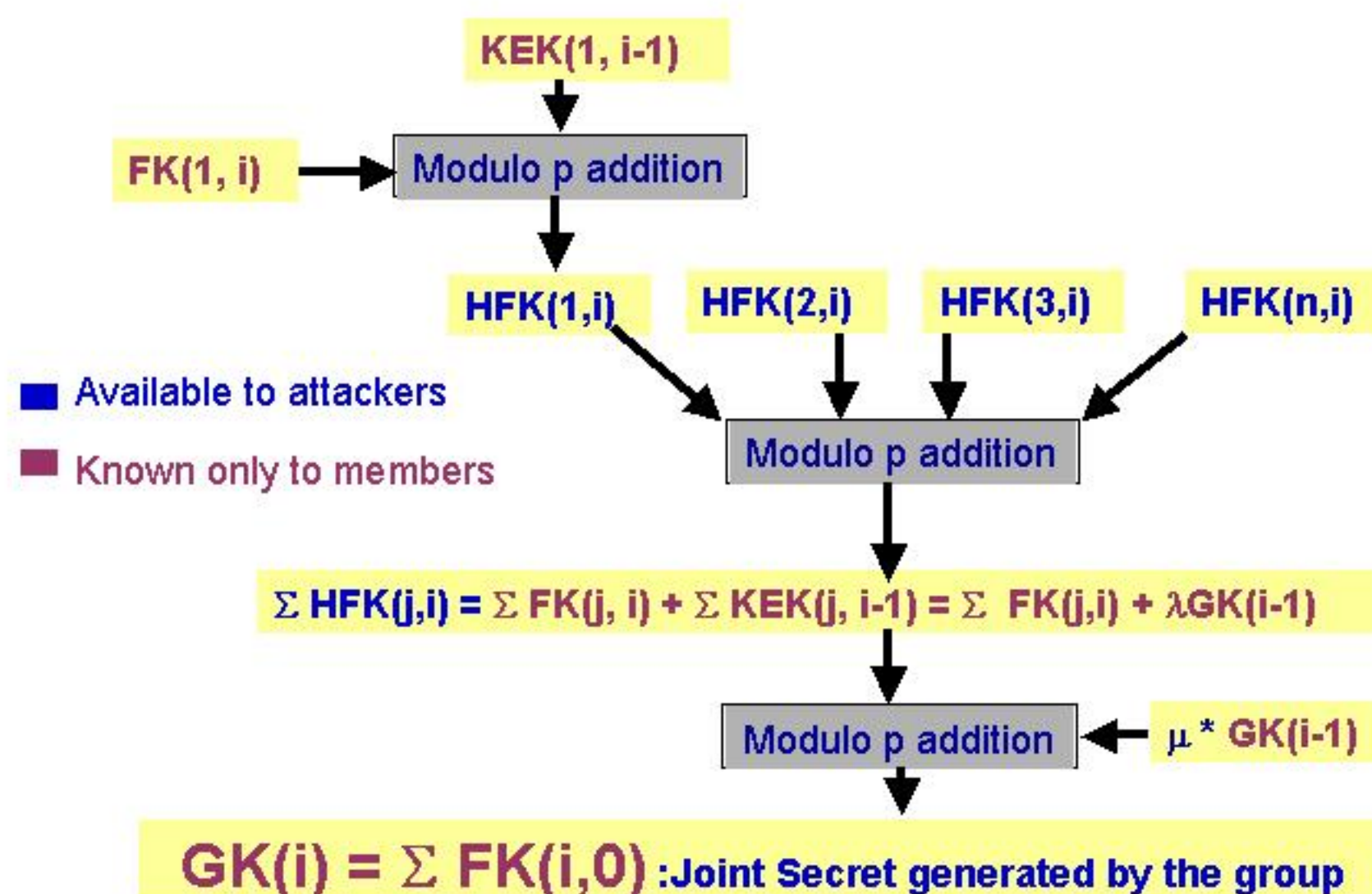


# Distributed Authenticated Key Generation for Mobile Ad hoc Networks

K. Manousakis, V. Bharadwaj/J. Baras

## Introduction

- **Assumptions**
  - Mobile ad hoc network - no fixed nodes or links
  - Many-to-many group communication using multicasting
  - Nodes have only partial trust in each other
  - No trusted third party (except maybe during initialization)
  - Nodes are capable of generating uniformly distributed Fractional Keys
  - Nodes are capable of masking their Fractional Keys to form Hidden Fractional Keys
- **Problem**
  - Implement a provably secure distributed key generation scheme to generate and distribute a shared key for the group without revealing individual secrets
- **Design Goals**
  - No time stamps or time synchronization between nodes
  - Minimize possibility of using Denial of Service (DoS) attacks
  - Avoid unnecessary encryptions
  - Minimize use of Public-key based computation
  - Minimize computation and communication overhead
- **Achievements**
  - Developed a contributory key generation protocol that is collusion-free and does not require exponentiation
  - Developed a version of the above protocol with distributed authentication
  - Developed Auxiliary Key Agreement (AKA) protocol to improve efficiency under dynamic group membership
  - Developed optimized version of AKA using Logical Key Trees
  - Proved protocol security using BAN logic



## Contributory Key Generation

- **Adversary model**
  - Passive attacker who can observe all messages but not create or alter messages
- **Features**
  - Computationally efficient
  - Collusion-free: member's fractional key is secret unless all the others collude against him
- **Drawbacks**
  - Vulnerable to active attack - HFKs are not authenticated
  - Inefficient for membership change - entire protocol must be rerun when a member joins or leaves
- **Protocol description**
  - Initialization:
    - » Group Initiator distributes *One Time Pads* to group members. These are the *Key Encrypting Keys* for the first round
    - » Initiator also reveals the *sum of all the One Time Pads* to each member. This is the *Group Key* for the first round.
  - Key Generation:
    - » Each member generates a *Fractional Key*.
    - » The *Fractional Key* is encrypted with the *Key Encrypting Key* to obtain a *Hidden Fractional Key*.
    - » The members announce their *Hidden Fractional Keys*.
    - » Each member combines the *Hidden Fractional Keys* and removes the combined effect of the *Key Encrypting Keys* to obtain the new *Group Key*. Thus the *Group Key* is the sum of the *Fractional Keys*.
    - » Each member obtains their new *Key Encrypting Key* by removing the effect of their *Fractional Key* from the new *Group Key*

## Authenticated Contributory Key Generation

- **Adversary model**
  - Active attacker who can observe all messages, and insert, drop or modify protocol messages
- **Additional Assumptions**
  - Public Key Infrastructure exists
  - Attacker cannot break one-way hash or compromise PKI
- **Protocol Guarantees**
  - External attacker cannot lead group members into forming an incorrect key. In other words the protocol provides key authentication.
  - External attackers cannot cause denial of service in a broadcast medium assuming that he cannot control the link layer.
- **Features**
  - Centralized version is collusion-free. Distributed version is collusion-resistant (a minimum number of people must collude to be successful.)
- **Protocol description**
  - Members first mutually authenticate and establish shared keys using PKI
  - Initialization:
    - » One-time pads are encrypted, then concatenated with an SKEY-based nonce and digitally signed.
  - Key Generation:
    - » In place of a Key Encrypting Key each member has a binding parameter
    - » The HFK messages also include a digital signature, which includes the binding parameter to ensure freshness

## Auxiliary Key Agreement

- **Features**
  - Member join:
    - » Centralized version: A random subset of members is given new *One Time Pads*. The remaining members are given the new binding parameter.
    - » Distributed version: New member finds a "sponsor" to help him generate a new *One Time Pad* and to update the binding parameters.
  - Member Revocation:
    - » Centralized version: Same as member join.
    - » Distributed version: Rerun authenticated key generation starting at member  $M_{i+1}$
- **Enhancements for Efficiency**
  - Centralized version can be made more efficient for satellite environments
    - » Use a Logical Key Tree for protocol messages
    - » MAC keys precomputed and distributed by Security Manager

