Outline

- A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. **SPINS: Security protocols for sensor networks**. In Proceedings of MOBICOM, 2001
  - Sensor networks and security are important
  - Sensors are deployed in hostile territory. The communications are sparse because of energy constraints, the computational resources are sparse. However, attackers need not be energy constrained, they can replay packets, inject spurious packets etc. and affect the system
    - How do we make security (heavy weight) fit into sensor scenarios (low resources)
    - Slides courtesy Adrian Perrig

Security in sensor networks

- Emergency response: responders need security
- Medical monitoring: automated drug delivery - need security to ensure safety
- Logistics and inventory management:
- Battlefield management

- Security:
  - Authentication
    - Ensures data integrity & origin
    - Prevents injecting bogus messages
  - Confidentiality
    - Ensures secrecy of data
    - Prevents eavesdropping

Challenges

- Integrity of sensor: hard to manage without expensive crypto processors or ensuring physical security
- Key distribution is a challenge
  - Don’t want to store private keys in sensors
  - Key strength weakens with time
- Freshness important
  - Prevent replay attack
  - Define notions of strong freshness (delay estimation, total ordering) and weak freshness (partial ordering)
- Keys are too long to store, much less process
- Authenticated broadcast challenging

Challenge: Resource Constraints

- Limited energy
- Limited computation (4 MHz 8-bit)
- Limited memory (512 bytes)
- Limited code size (8 Kbytes)
  - ~3.5 K base code ("TinyOS" + radio encoder)
  - Only 4.5 K for application & security
- Limited communication (30 byte packets)
- Energy-consuming communication
  - 1 byte transmission = 11000 instructions

SPINS: Our Solution

- SNEP
  - Sensor-Network Encryption Protocol
  - Secures point-to-point communication
- μTESLA
  - Micro Timed Efficient Stream Loss-tolerant Authentication
  - Provides broadcast authentication

System Assumptions

- Communication patterns
  - Frequent node-base station exchanges
  - Frequent network flooding from base
  - Node-node interactions infrequent
- Base station
  - Sufficient memory, power
  - Shares secret key with each node
- Node
  - Limited resources, limited trust

SNEP Security Goals

- Secure point-to-point communication
  - Confidentiality, secrecy
  - Authenticity and integrity
  - Message freshness to prevent replay
- Why not use existing protocols?
  - E.g. SSL/TLS, IPSEC

Encryption methods (background)

- Symmetric cryptography
  - Sender and receiver know the secret key (apriori)
    - Fast encryption, but key exchange should happen outside the system

- Asymmetric cryptography
  - Each person maintains two keys, public and private
    - M ≡ PrivateKey(PublicKey(M))
    - M ≡ PublicKey (PrivateKey(M))
  - Public part is available to anyone, private part is only known to the sender
  - E.g. Pretty Good Privacy (PGP), RSA

Asymmetric Cryptography is Unsuitable

- Overhead of digital signatures
  - High generation cost O(minutes)
  - High verification cost O(seconds)
  - High memory requirement
  - High communication cost ~128 bytes
- SNEP only uses symmetric crypto

Basic Crypto Primitives

- Code size constraints ⇒ code reuse
- Only use block cipher encrypt function
  - Counter mode encryption
  - Cipher-block-chaining message authentication code (MAC)
  - Pseudo-Random Generator

SNEP Protocol Details

- A and B share
  - Encryption keys: K_{AB} K_{BA}
  - MAC keys: K'_{AB} K'_{BA}
  - Counters: C_A C_B
- To send data D, A sends to B:

$A \rightarrow B$: $\{D\}_{<K_{AB}, C_A>}$

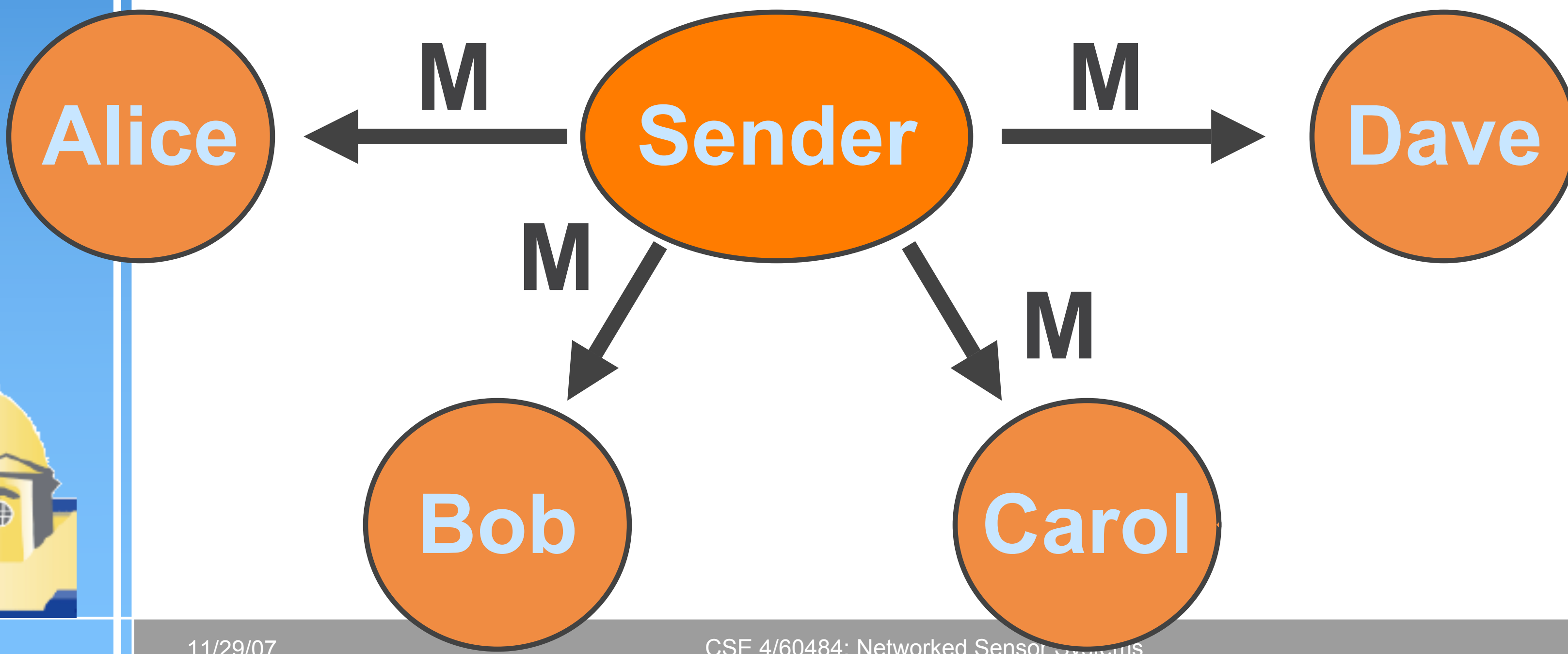
$MAC(K'_{AB}, [C_A \,||\, \{D\}_{<K_{AB}, C_A>}])$

SNEP Properties

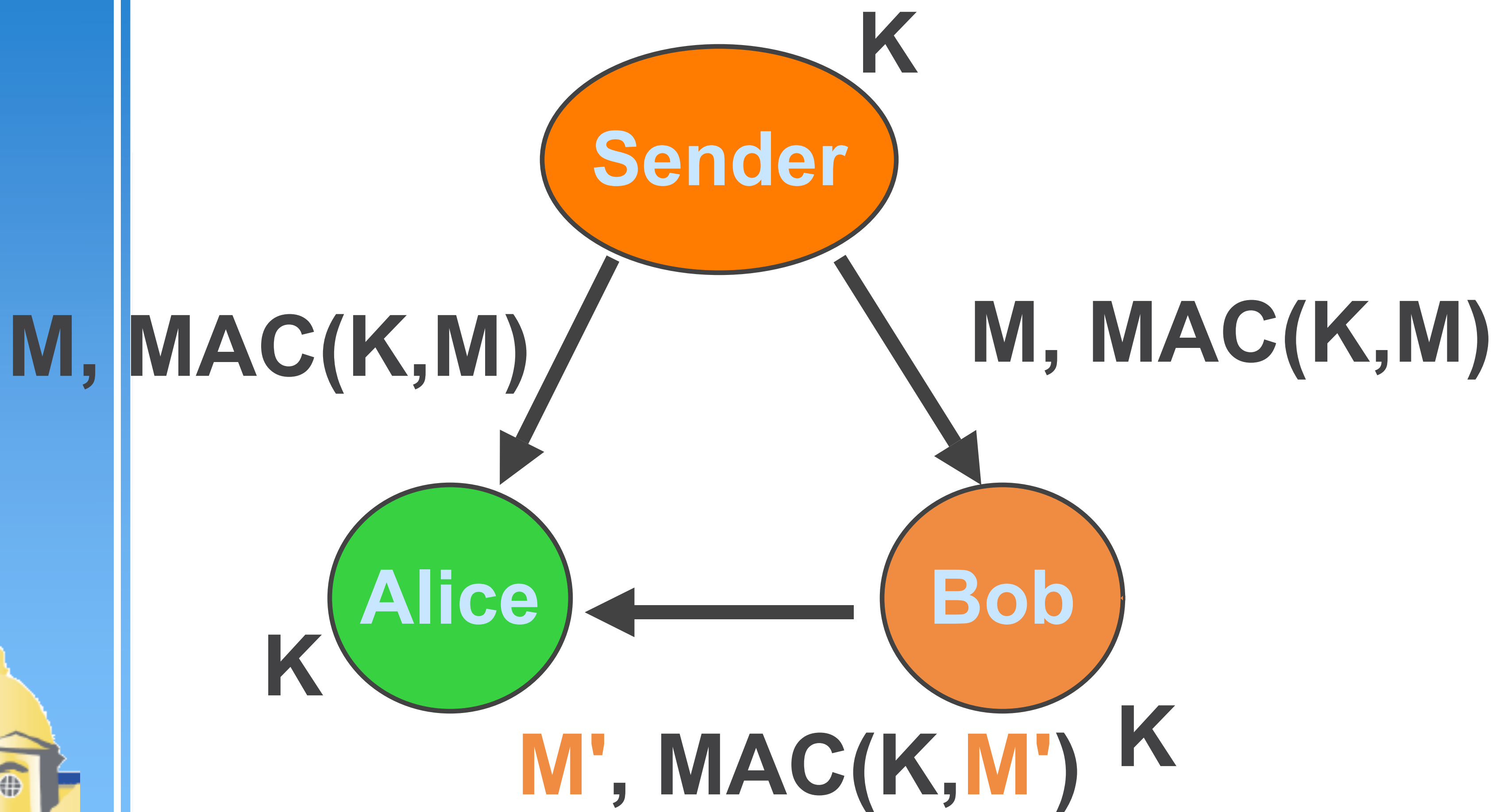
- Secrecy & confidentiality
  - Semantic security against chosen ciphertext attack (strongest security notion for encryption)
- Authentication
- Replay protection
- Code size: 1.5 Kbytes
- Strong freshness protocol in paper

Broadcast Authentication

- Broadcast is basic communication mechanism
- Sender broadcasts data
- Each receiver verifies data origin



Simple MAC Insecure for Broadcast



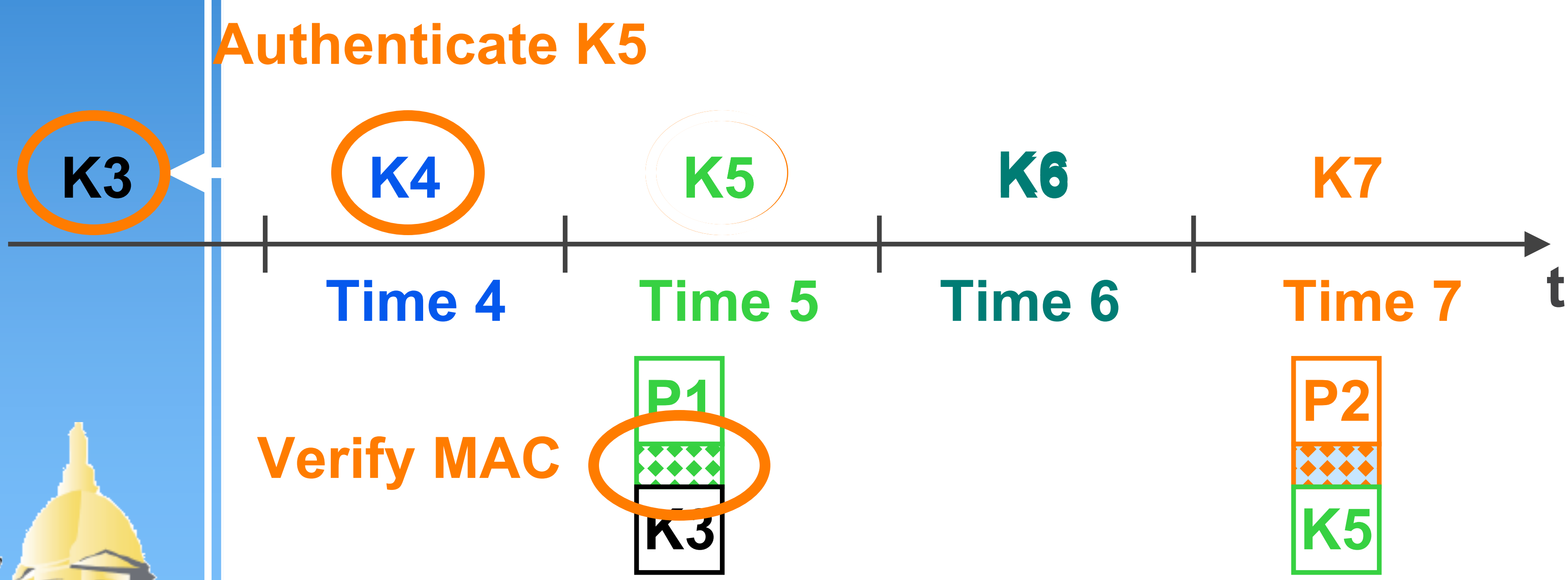
μTESLA: Authenticated Broadcast

- Uses purely symmetric primitives
- Asymmetry from delayed key disclosure
- Self-authenticating keys
- Requires loose time synchronization
  - Use SNEP with strong freshness

μTESLA Quick Overview I

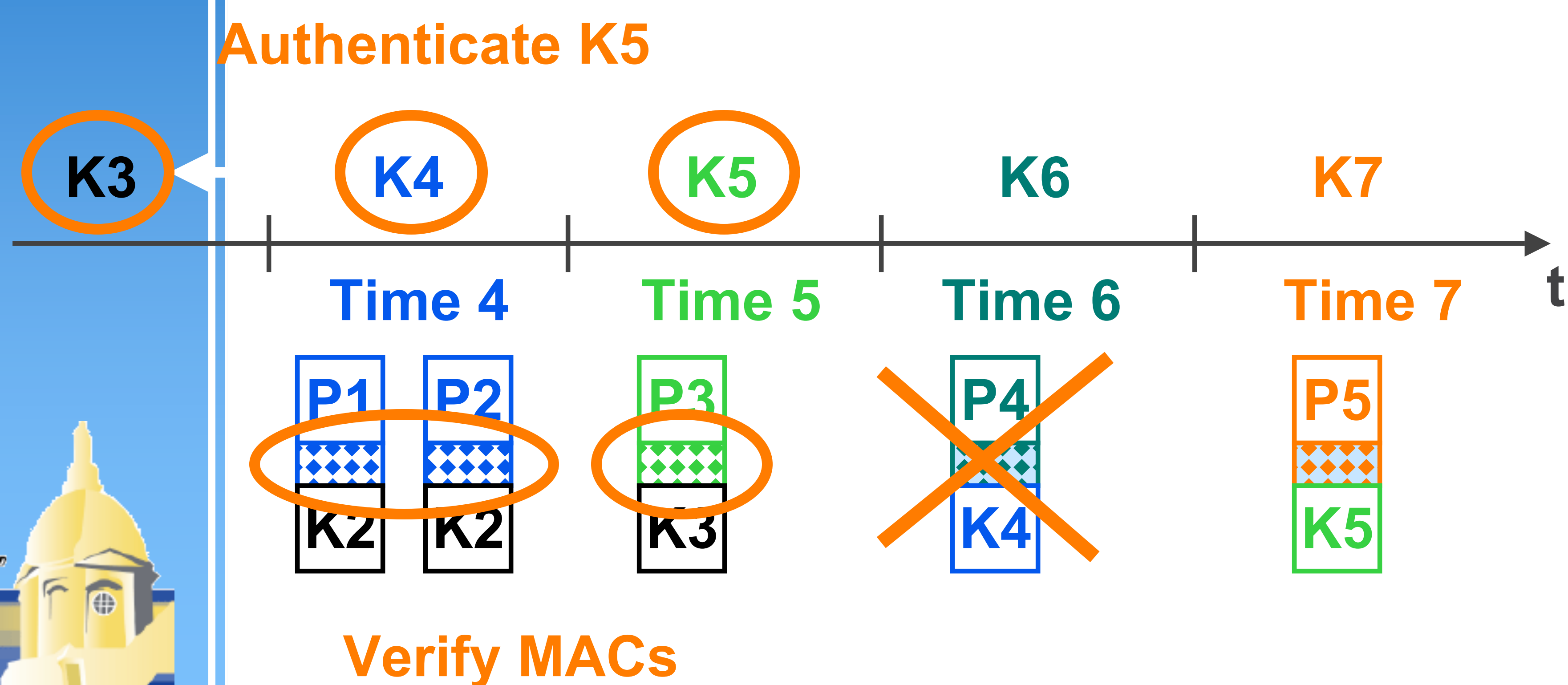
- Keys disclosed 2 time intervals after use
- Receiver knows authentic K3

- **Authentication of P1: MAC(K5, P1)**



µTESLA Quick Overview II

- Perfect robustness to packet loss

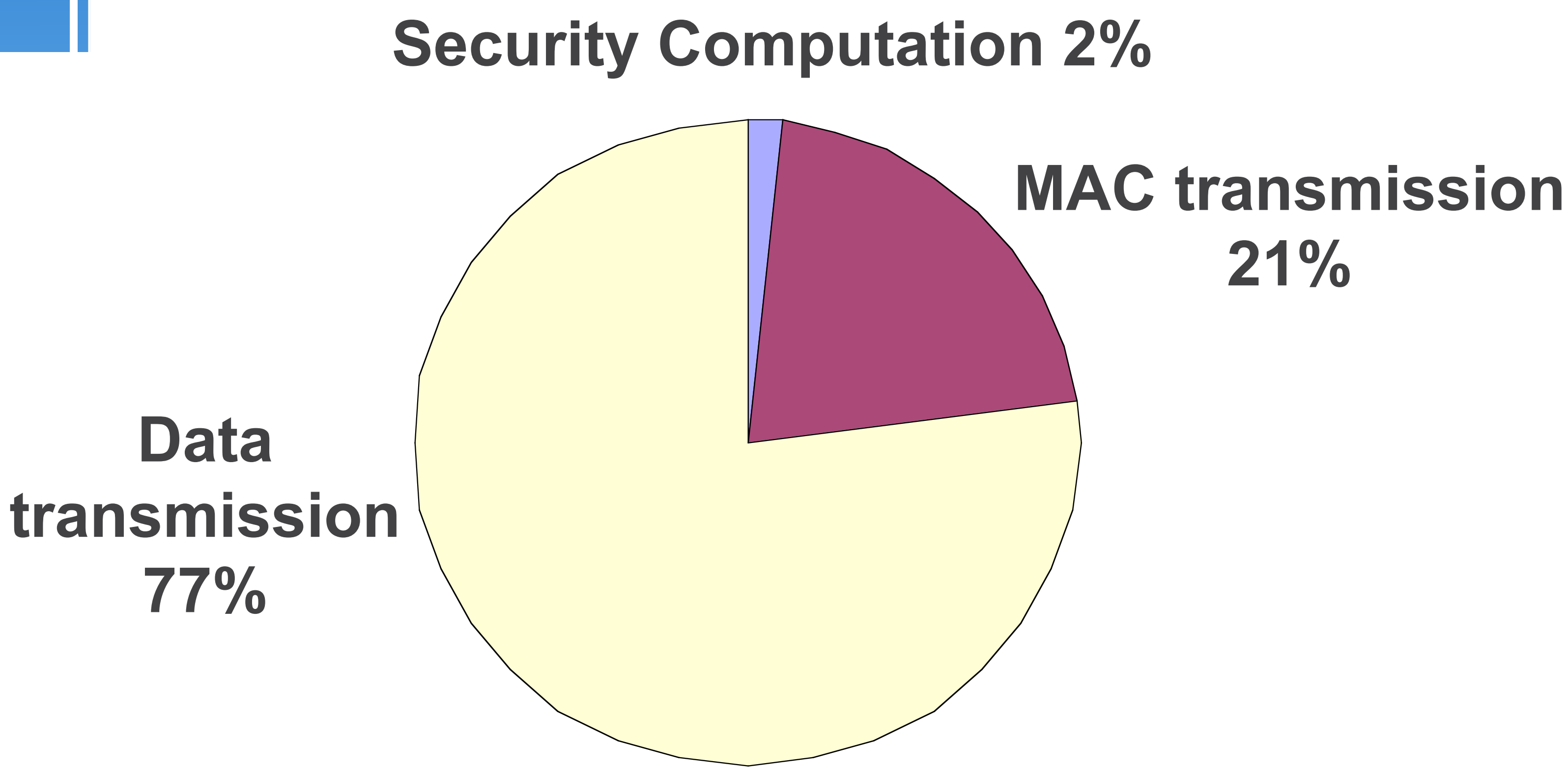


μTESLA Properties

- Low overhead (1 MAC)
  - Communication (same as SNEP)
  - Computation (~ 2 MAC computations)
- Perfect robustness to packet loss
- Independent of number of receivers

Energy Cost for Sending a Message

- Typical packet size: 28 bytes



Conclusion

- Strong security protocols affordable
  - First broadcast authentication
- Low security overhead
  - Computation, memory, communication
- Apply to future sensor networks
  - Energy limitations persist
  - Tendency to use minimal hardware
- Base protocol for more sophisticated security services