A Symmetric Key Generation System (KGS) Suitable for Sensor/Building Networks

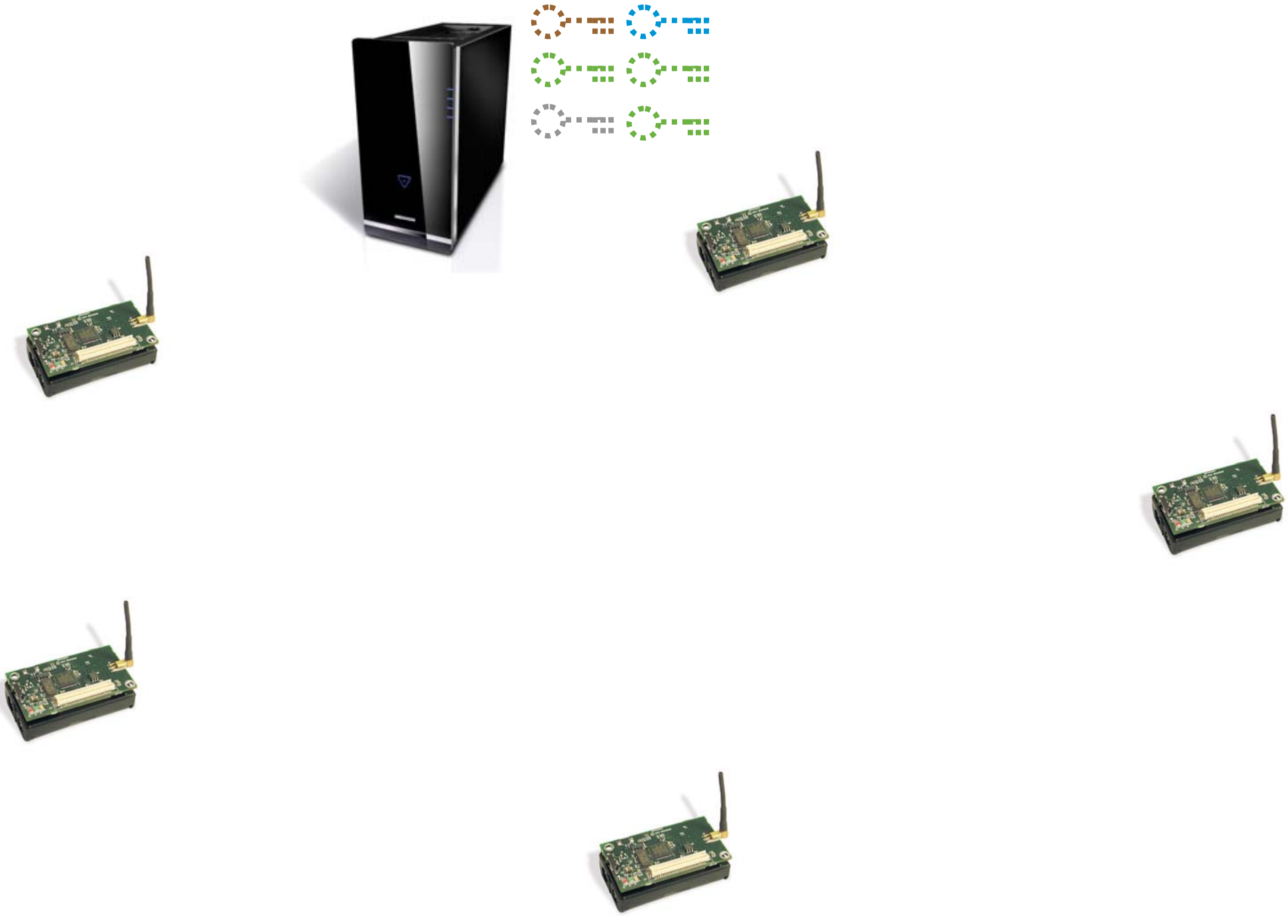
David McGrew, Cisco Fellow, mcgrew@cisco.com

Brian Weis, Cisco Distinguished Engineer, bew@cisco.com

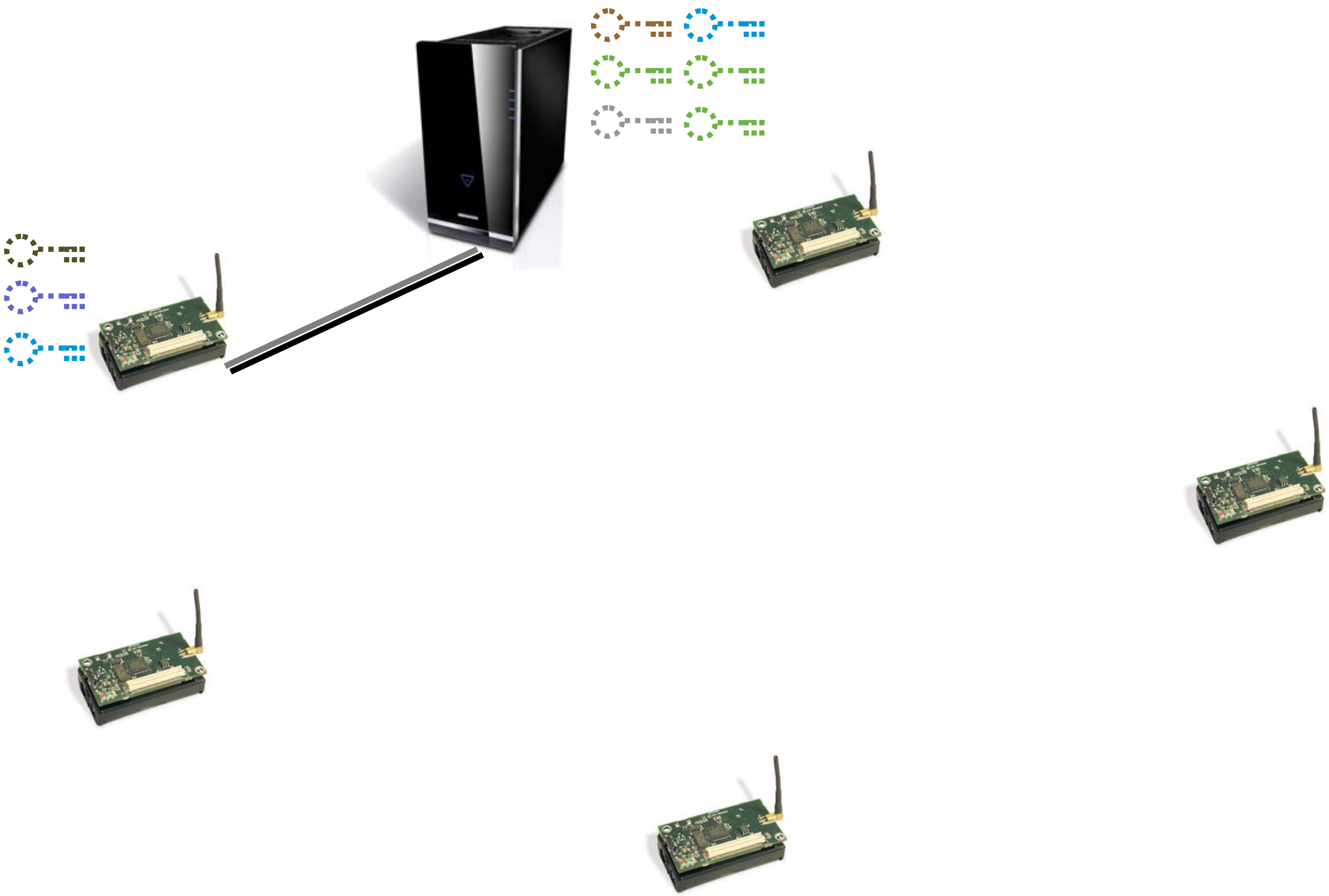
KGS Goals

- Key management for devices with limited computation and communication
  - Low power wireless
- Allow full or partial mesh communication
- Access control defined by group controller
- Security
  - Even if multiple devices are compromised
  - Even if communication keys are leaked

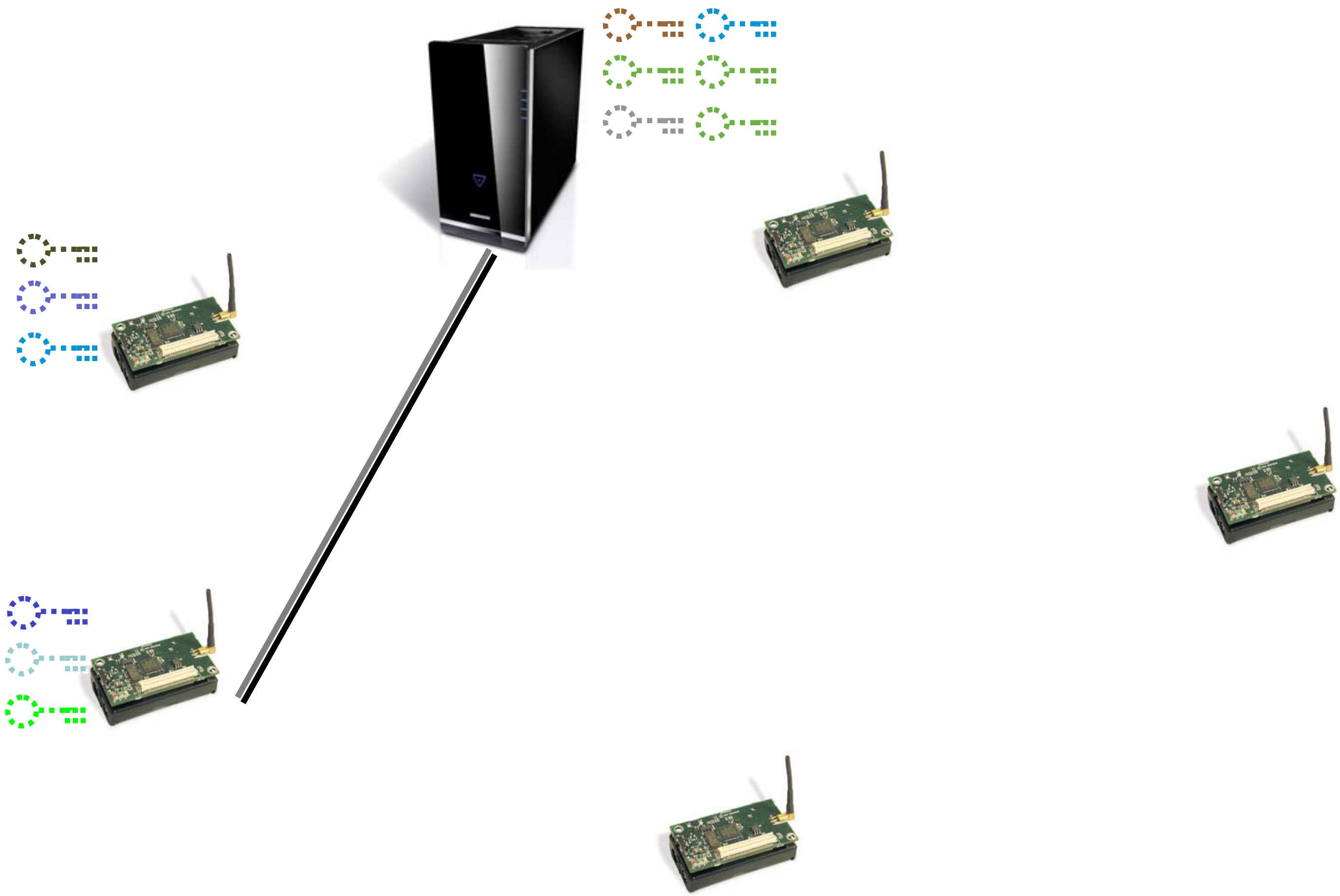
Controller Initialization



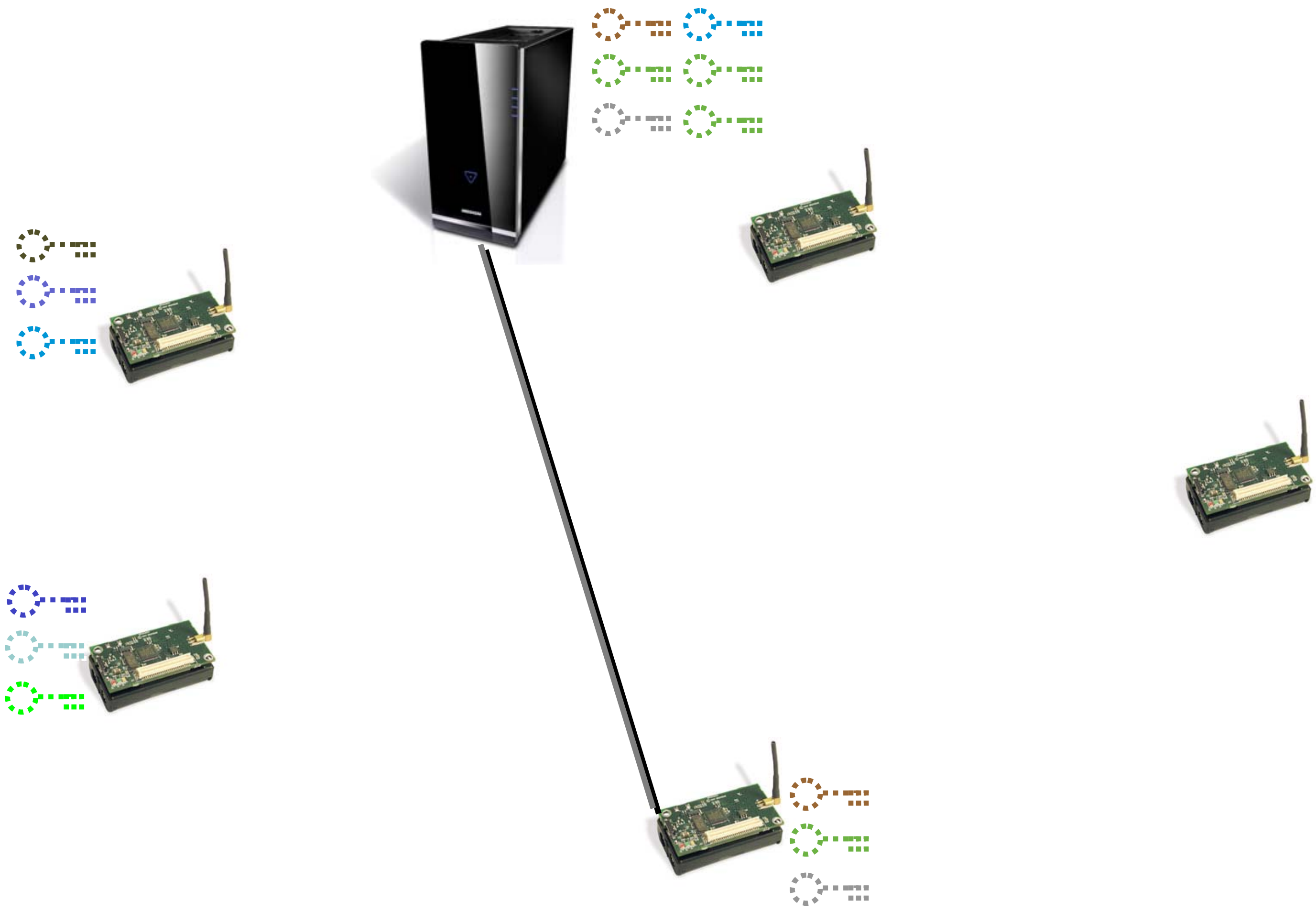
User Initialization



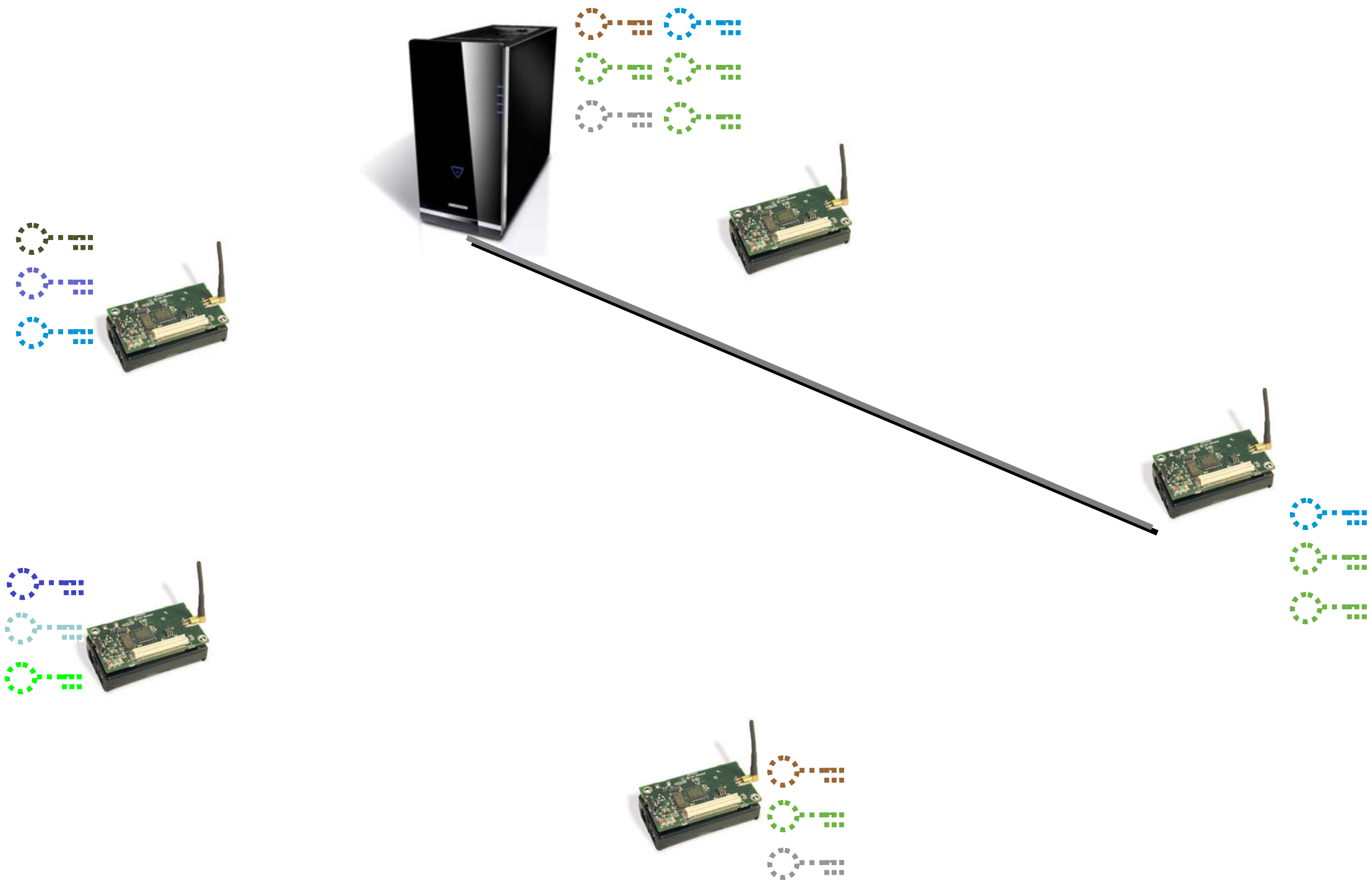
User Initialization



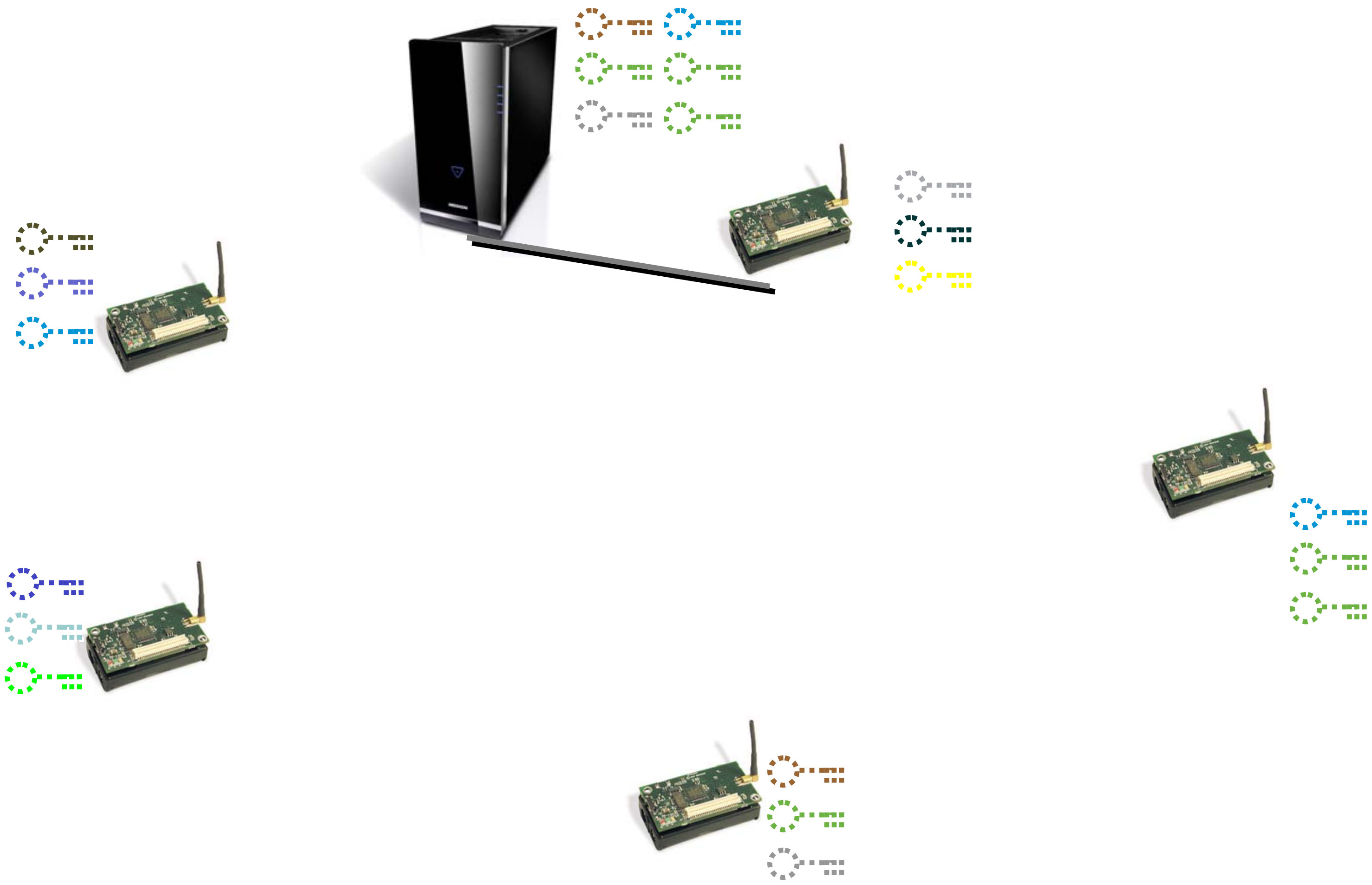
User Initialization



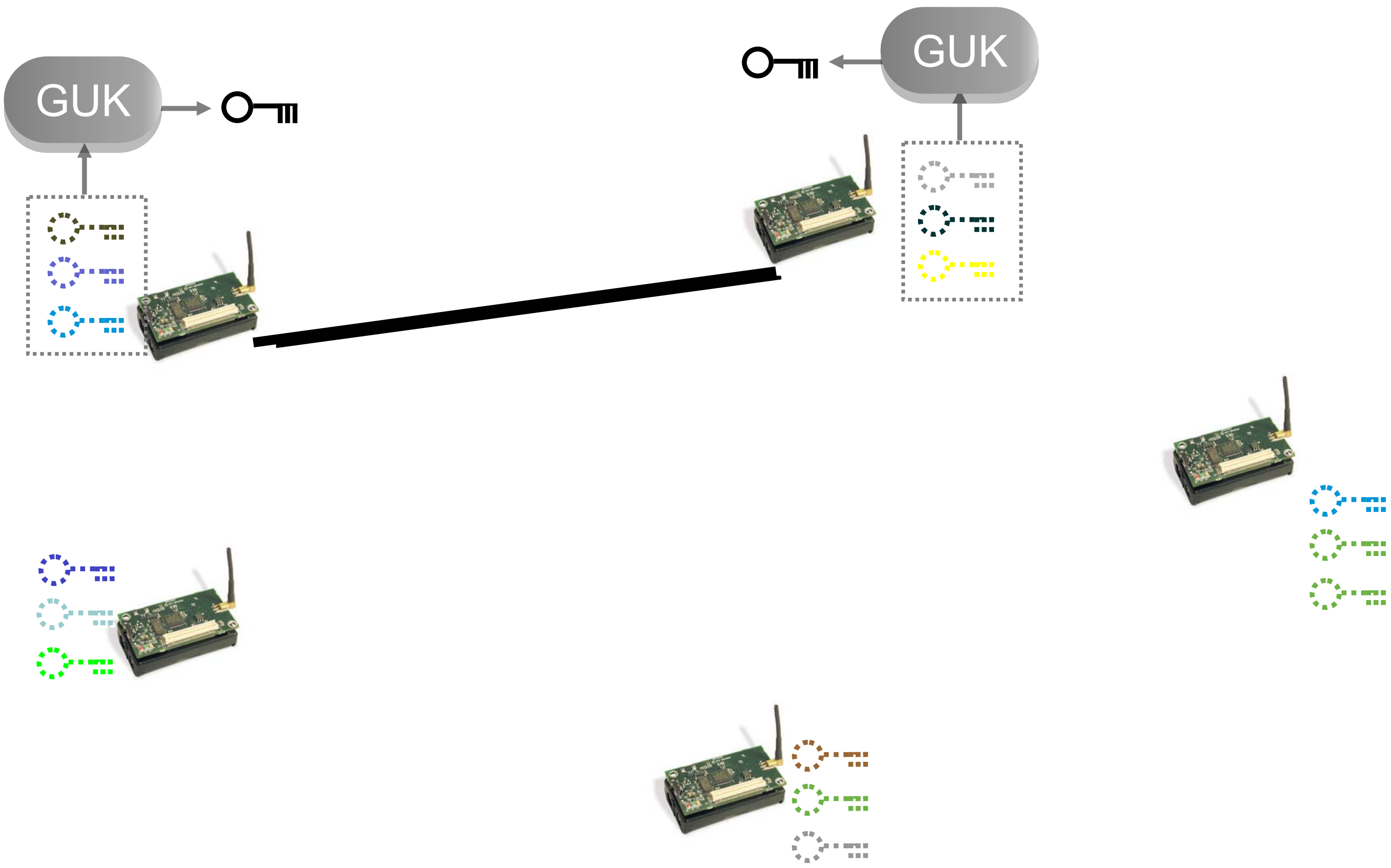
User Initialization



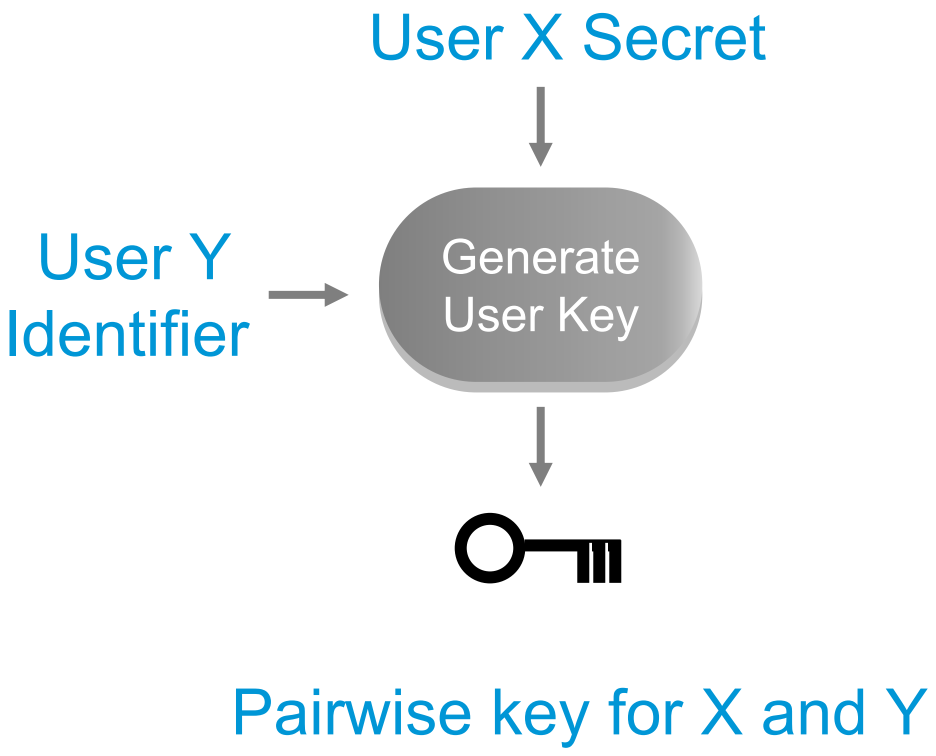
User Initialization



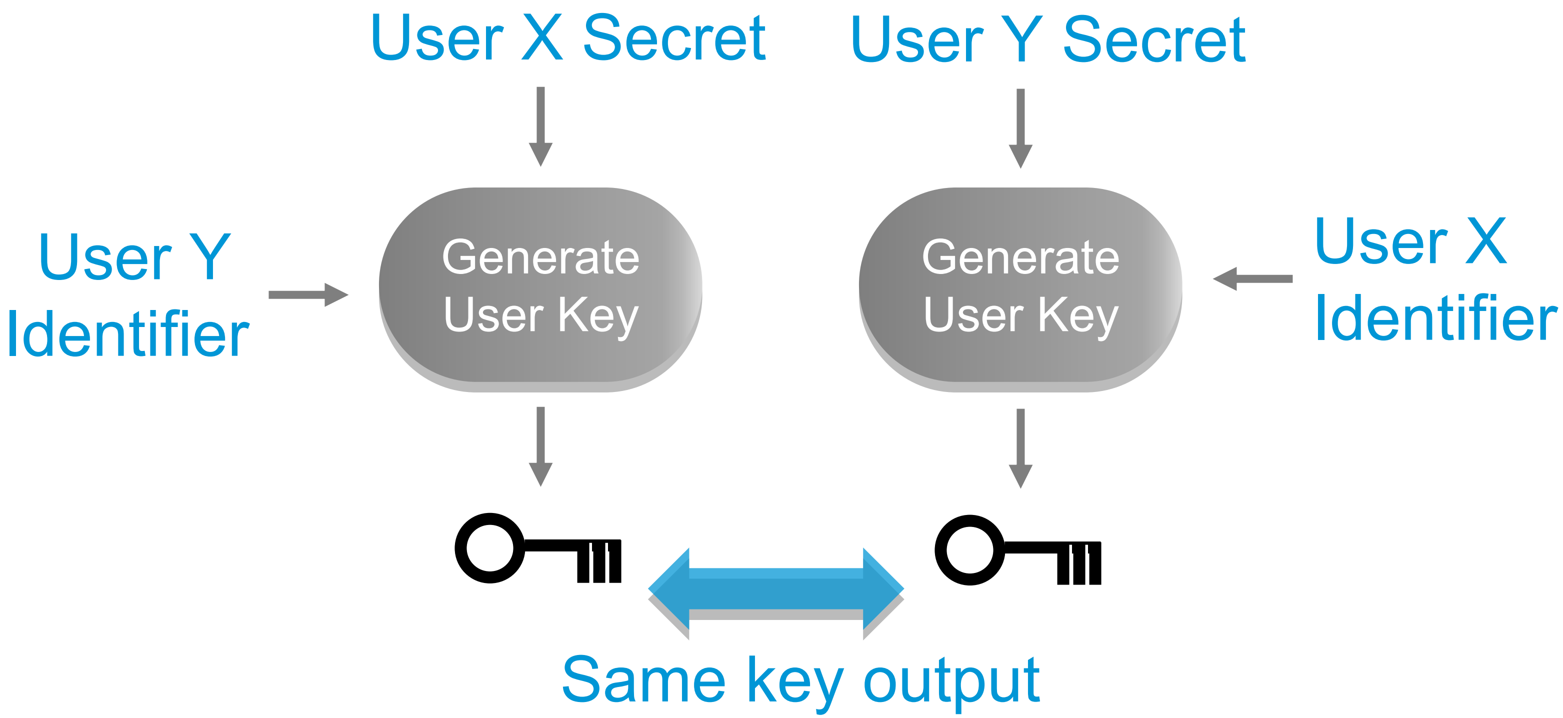
Pairwise key generation



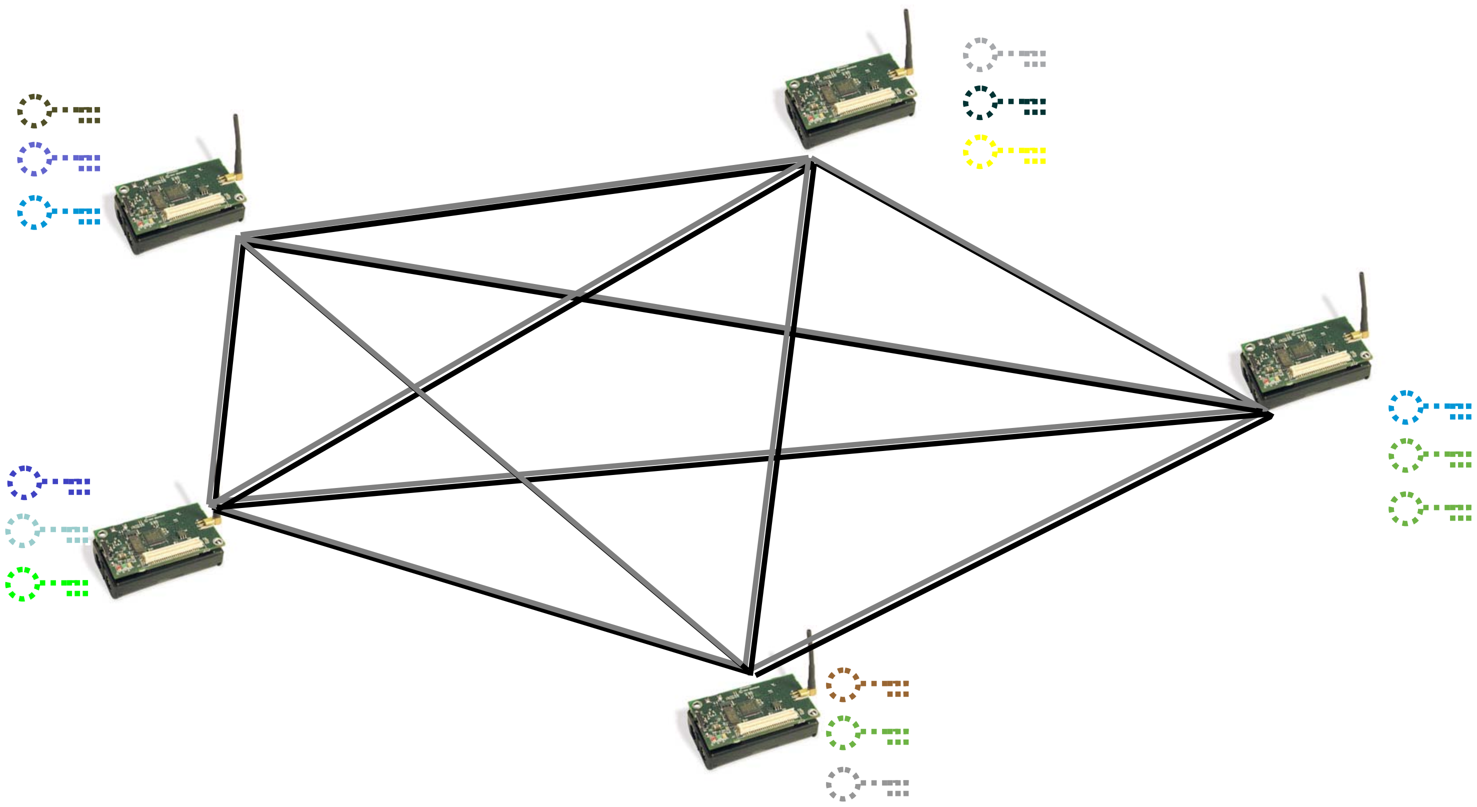
Key generation function



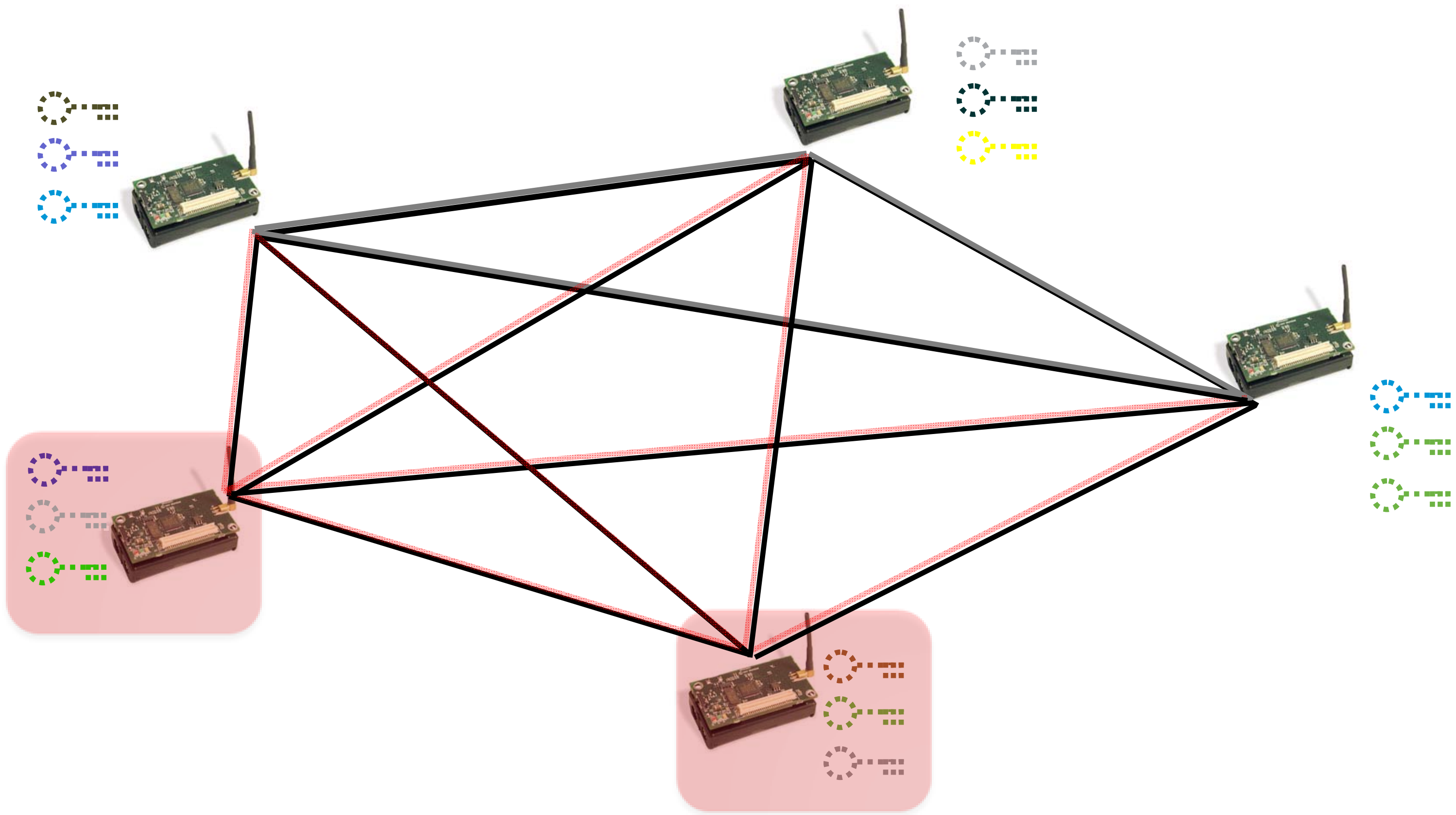
Key generation function



Pairwise key generation



T-compromise resistance



Blundo et al '96 (Symmetric Polynomial)

Controller Initialization $D_{ij} = D_{ji} = \text{rand}$

$K_{xy} = \Sigma_i \Sigma_j D_{ij} x^i y^j$

Controller Add User $U^x{}_i = \Sigma_j D_{ij} x^j$

User Generate Key $K_{xy} = \Sigma_i U^x{}_i y^i$

Information theoretic security

- For a KGS with a threshold t and c corrupted users,

  If $c < t$, then the adversary has no information about the keys generated by the system, assuming a perfect random source.

  If $c < t$ and the random source has a statistical distance from random of at most ϵ, then the adversary has advantage at most ϵ^{t-c}.

Concrete Security Model

- Related Keys

  For any set of keys {k_1 , k_2 , . . . , k_l } generated by the KGS with a threshold of t, with l > s = (t + 1)(t + 2)/2, such that the identifier-pair associated with each key is distinct, there exist coefficients c_1 , c_2 , . . . , c_s □ GF (q) such that $\Sigma_{i=1,s} c_i \cdot k_i = 0$. The values of these coefficients can be determined from the identifier-pairs.

- Revocation and Forward Security

$$K'_{xy} = \mathrm{KDF}(K_{xy}, E)$$

Epoch Parameter

Computations

$$K_{xy}=(((U^{x}_{k}\,y + U^{x}_{k-1})y + U^{x}_{k-2})y + \ldots + U^{x}_{0})$$

Computations

$$K_{xy}=(((U^x_k\, y + U^x_{k-1})y + U^x_{k-2})y + \ldots + U^x_0)$$

$$X=((((A_0H + A_1)H + A_2)H + \ldots + A_k)H + L)H$$

KGS	GCM
User *X*'s array *U*	Authenticated data *A*
User identifier *Y*	Hash key *H*

KGS using $GF(2^{128})$

- GCM field $GF(2^{128})$
  - 128-bit security level
  - Allows up to $2^{128} - 1$ group members
  - User stores $16(t+1)$ bytes
  - Controller stores $8(t+1)(t+2)$ bytes
- Pairwise key generation
  - Essentially the same as processing $16t$ bytes with AES-GMAC
  - $t \sim 90$ is equivalent to a typical packet size (1440 bytes)
- Key generation as fast as data plane
  - Pairwise keys can be computed on demand

Pseudorandom *D*-array

- *D*-array takes $O(t^2)$ random input, $O(t^2)$ storage

Random $$D_{ij} = D_{ji} = \mathrm{rand}$$

Pseudoandom $$D_{ij} = D_{ji} = \mathrm{KDF}(i \,||\, j)$$

Standards

- KDF for generating *D*-array
  - SP 800 108
- KDF for post-processing
  - SP 800 108
- GF(*q*) computations
  - SP 800 38 D
- Distribution of User Secret
  - EAP / 802.1X
  - GDOI
  - GSAKMP

Conclusions

- Blundo et al KGS is practical
  - Trivial to implement, given GCM primitives
  - Re-use of components is ideal for sensor nodes
- Security is strong and well-understood
  - Information theoretic and concrete security models
- Can extend functionality of NIST cryptographic toolkit

Thank you.