Techniques for Key Predistribution in Networks

Douglas R. Stinson
David R. Cheriton School of Computer Science
University of Waterloo

UNC Charlotte
April 11, 2014

Outline

1. Introduction to key predistribution schemes (KPS)
2. The Blom Scheme
3. KPS for sensor networks: random and deterministic schemes
4. ID-based one-way function schemes for specified communication graphs

Key Predistribution Schemes

- Protocols enabling secure distribution of keys are of fundamental importance in cryptography.
- These include **key agreement**, **key transport** and **key predistribution**.
- In this talk, we focus on key predistribution schemes (KPS), where a trusted authority (the TA) distributes secret information securely to the network users before the network is deployed.
- The keys are symmetric pairwise keys for use in secret-key crypto systems (e.g., AES).
- As an example, the nodes in a **wireless sensor network** may be preloaded with keys or keying information, and then nodes may be scattered from an airplane (random deployment).

Attack Models and Adversarial Goals for KPS

- We consider a network of n nodes which do not necessarily trust each other.
- An adversary may corrupt a subset of the nodes, and obtain all their secret information.
- We sometimes consider adversaries who corrupt up to κ nodes, where κ is a **security parameter**.
- The adversary's goal is to determine the secret key corresponding to a pair of uncorrupted nodes.
- We first describe the *Blom KPS*, which is a KPS that is **unconditionally secure** (AKA **information-theoretically secure**) against adversaries of this type.
- This means that the security can be proven mathematically without making any computational assumptions.

Two Trivial Schemes

1. If every node is given the same secret master key, then memory costs are low. However, this situation is unsuitable because the compromise of a single node would render the network completely insecure.

2. For every pair of nodes, there could be a secret pairwise key given only to these two nodes. This scheme would have optimal resilience to node compromise, but memory costs would be prohibitively expensive for large networks because every node would have to store $n - 1$ keys, where n is the number of nodes in the network.

The Blom KPS (1982)

Here is the Blom scheme for $\kappa = 1$. For each node U, a value $r_U \in \mathbb{Z}_p$ is made public (where $p \geq n$ is prime). The values r_U are distinct elements of $\mathbb{Z}_p$.

Protocol: *Blom's key distribution scheme ($\kappa = 1$)*

1. The TA chooses three random elements $a, b, c \in \mathbb{Z}_p$ (not necessarily distinct), and forms the polynomial

$$f(x, y) = a + b(x + y) + cxy \bmod p.$$

2. For each node U, the TA computes the polynomial

$$g_U(x) = f(x, r_U) \bmod p = a_U + b_U x$$

and transmits (a_U, b_U) to U over a secure channel.

The Blom PKS (cont.)

- The key for U and V is

$$K_{U,V} = K_{V,U} = f(r_U, r_V),$$

  where U computes $K_{U,V} = g_U(r_V)$ and V computes $K_{U,V} = g_V(r_U)$.
- We have:

$$\begin{aligned} a_U &= a + b\,r_U \bmod p \quad \text{and} \\ b_U &= b + c\,r_U \bmod p, \quad \text{so} \\ g_U(r_V) &= a + b\,r_U + (b + c\,r_U)r_V \\ &= a + b(r_U + r_V) + c\,r_U r_V \\ &= f(r_U, r_V) \bmod p. \end{aligned}$$

A Toy Example

- Suppose $p = 17$.
- Suppose there are three nodes: U, V and W, and their public values are $r_U = 12$, $r_V = 7$ and $r_W = 1$.
- Suppose the TA chooses $a = 8$, $b = 7$ and $c = 2$, so the polynomial f is

$$f(x, y) = 8 + 7(x + y) + 2xy.$$

- The g polynomials are as follows:

$$\begin{aligned} g_U(x) &= 7 + 14x \\ g_V(x) &= 6 + 4x \\ g_W(x) &= 15 + 9x. \end{aligned}$$

A Toy Example

- The three keys are

$$\begin{aligned} K_{U,V} &= 3 \\ K_{U,W} &= 4 \\ K_{V,W} &= 10. \end{aligned}$$

- U would compute $K_{U,V}$ as

$$g_U(r_V) = 7 + 14 \times 7 \bmod 17 = 3.$$

- V would compute $K_{U,V}$ as

$$g_V(r_U) = 6 + 4 \times 12 \bmod 17 = 3.$$

Unconditional Security of the Blom Scheme ($\kappa = 1$)

- We show that no individual node, say W, can determine any information about a pairwise key for two other nodes, say $K_{U,V}$.
- What information does W possess?
- W knows the values

$$a_W = a + b\, r_W \bmod p$$

and

$$b_W = b + c\, r_W \bmod p.$$

- The key that W is trying to compute is

$$K_{U,V} = a + b(r_U + r_V) + c\, r_U r_V \bmod p.$$

- The values r_U, r_V and r_W are public, but a, b and c are unknown.

Security of the Blom Scheme (cont.)

- We will show that the information known by W is consistent with any possible value $K^* \in \mathbb{Z}_p$ of the key $K_{U,V}$.
- Consider the following matrix equation (in $\mathbb{Z}_p$):

$$\begin{pmatrix} 1 & r_U + r_V & r_U r_V \\ 1 & r_W & 0 \\ 0 & 1 & r_W \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} K^* \\ a_W \\ b_W \end{pmatrix}.$$

- The determinant of the coefficient matrix is

$$r_W{}^2 + r_U r_V - (r_U + r_V) r_W = (r_W - r_U)(r_W - r_V),$$

  where all arithmetic is done in $\mathbb{Z}_p$.
- Since $r_W \neq r_U$ and $r_W \neq r_V$, it follows that the coefficient matrix has non-zero determinant, and hence the matrix equation has a unique solution for a, b and c.

Security of the Blom Scheme (cont.)

- However, a coalition of two nodes, say $\{W, X\}$, will be able to compute **any** key $K_{U,V}$ where $\{W, X\} \cap \{U, V\} = \emptyset$.
- W and X together have the following information:

$$\begin{aligned} a_W &= a + b\,r_W \\ b_W &= b + c\,r_W \\ a_X &= a + b\,r_X \\ b_X &= b + c\,r_X, \end{aligned}$$

  where a, b and c are unknowns.
- W and X together have **four** equations in **three** unknowns, and they can easily compute a unique solution for a, b and c.
- Once they know a, b and c, they can form the polynomial $f(x, y)$ and compute any key they wish.

The Blom Scheme for arbitrary κ

The general version of the the *Blom Scheme* uses a symmetric bivariate polynomial of degree k in each variable x and y.

The *Blom Scheme* with security parameter κ satisfies the following security properties:

1. no set of κ nodes, say $W_1, \ldots, W_\kappa$ can determine any information about a pairwise key for two other nodes, say $K_{U,V}$
2. any set of $\kappa + 1$ nodes, say $W_1, \ldots, W_{\kappa+1}$, can break the scheme

Fundamental Problems for WSNs

Eschenauer and Gligor (2002) introduced the following problems:

Key predistribution

We do not want to use a single key across the whole network due to the possibility of node compromise. So each node will receive a moderate sized key ring.

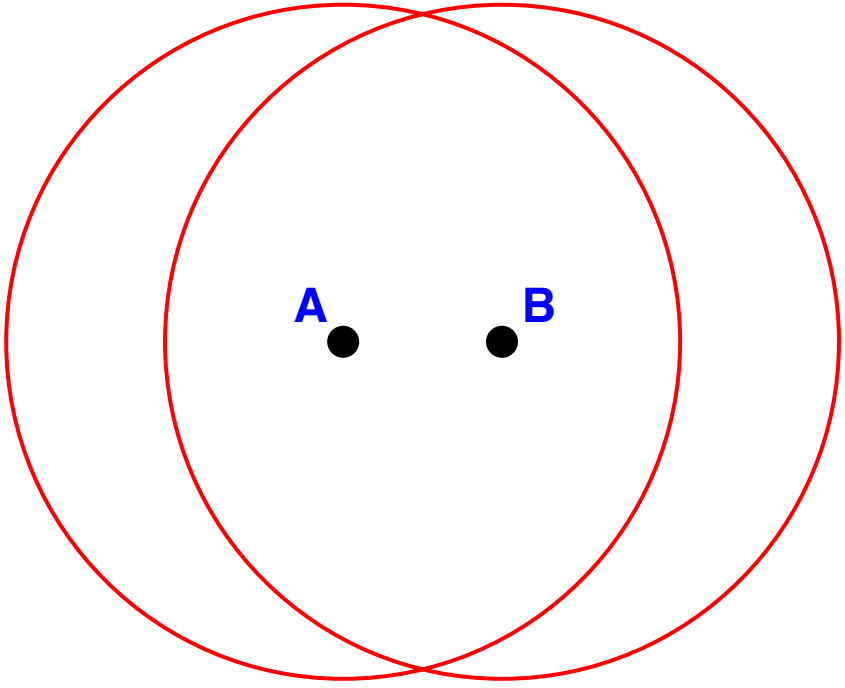
Shared-key discovery

Two nodes can communicate directly only if they are in close physical proximity and they have a common key. We need an efficient method to determine if two nearby nodes share a common key.

Path-key establishment

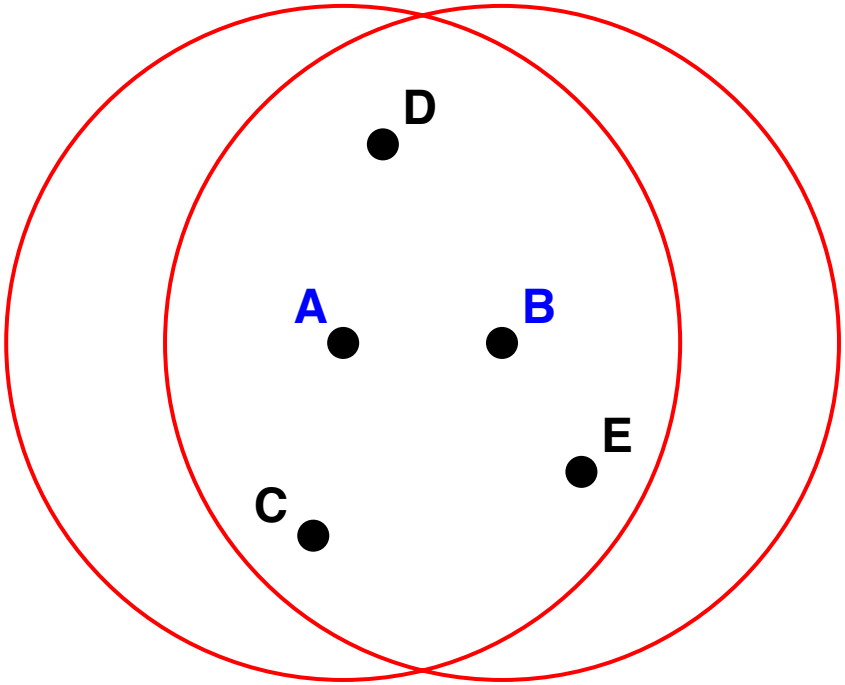
Nodes that cannot communicate directly should be able to communicate via a multi-hop path (preferably, a two-hop path). We need an efficient method for two nodes to determine a secure multi-hop path.

Shared-key Discovery



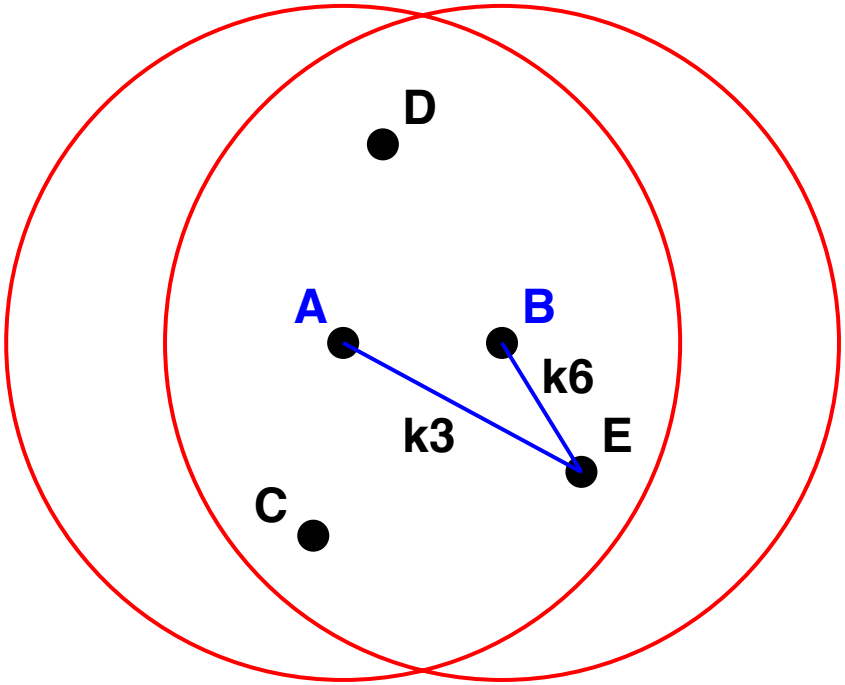
A has keys k1, k3, k5
B has keys k2, k4, k6

Path-key Establishment



A has keys k1, k3, k5
B has keys k2, k4, k6
C has keys k1, k3, k7
D has keys k2, k6, k7
E has keys k3, k6, k7

Path-key Establishment (cont.)



A has keys k1, k3, k5
B has keys k2, k4, k6
C has keys k1, k3, k7
D has keys k2, k6, k7
E has keys k3, k6, k7

The Eschenauer-Gligor Scheme

- In 2002, Eschenauer and Gligor proposed a randomized approach to key predistribution for sensor networks.
- For a suitable value of k, every node is assigned a random k-subset of keys chosen from a given pool of v secret keys.
- Suppose that nodes $\mathbf{N_i}$ and $\mathbf{N_j}$ have exactly $\ell \geq 1$ common keys, say $\mathbf{key}_{a_1}, \ldots, \mathbf{key}_{a_\ell}$, where $a_1 < a_2 < \cdots < a_\ell$ and $i < j$.
- Such a pair of nodes is termed an ℓ-link.
- Then $\mathbf{N_i}$ and $\mathbf{N_j}$ can each compute the same secret key,

$$K_{i,j} = h(\mathbf{key}_{a_1} \parallel \ldots \parallel \mathbf{key}_{a_\ell} \parallel i \parallel j),$$

  using a public key derivation function h.
- h could be constructed from a cryptographic hash function.

Attack Model: Random Node Compromise

- Suppose an adversary compromises a fixed number s of randomly chosen nodes in the network and extracts the keys stored in them.
- Any links involving the compromised nodes are broken.
- However, other links that do not directly involve the compromised nodes may also be broken.
- A link formed by two nodes $\mathbf{N_i}$ and $\mathbf{N_j}$, will be broken when a compromised node $\mathbf{N_k} \notin \{\mathbf{N_i}, \mathbf{N_j}\}$ contains all the keys held by $\mathbf{N_i}$ and $\mathbf{N_j}$, i.e., when $\mathbf{N_i} \cap \mathbf{N_j} \subseteq \mathbf{N_k}$.
- If s nodes, say $\mathbf{N_{k_1}}, \ldots, \mathbf{N_{k_s}}$, are compromised, then a link $\mathbf{N_i}, \mathbf{N_j}$ will be broken whenever

$$\mathbf{N_i} \cap \mathbf{N_j} \subseteq \bigcup_{h=1}^{s} \mathbf{N_{k_h}}.$$

Three Important Metrics

Storage requirements

The number of keys stored in each node, which is denoted by k, should be "small" (e.g., at most 100).

Network connectivity

The probability that a randomly chosen pair of nodes can compute a common key is denoted by $\mathbf{Pr_1}$. $\mathbf{Pr_1}$ should be "large" (e.g., at least 0.5).

Network resilience

The probability that a random link is broken by the compromise of s randomly chosen nodes not in the link is denoted by $\mathbf{fail_s}$. We want $\mathbf{fail_s}$ to be small: high resilience corresponds to a small value for $\mathbf{fail_s}$. In this talk we mostly consider $\mathbf{fail_1}$.

Deterministic Key Predistribution Schemes

- The Eschenauer-Gligor schemes are randomized schemes, in that the keys assigned to each node are chosen randomly.
- In 2004, deterministic KPS were proposed independently by Çamtepe and Yener; by Lee and Stinson; and by Wei and Wu.
- In a deterministic scheme, the assignment of keys to nodes is done in a deterministic fashion.
- A suitable set system (i.e., a design) is chosen, and each block is assigned to a node in the WSN (the design and the correspondence of nodes to blocks is public).
- The points in a block are the indices (i.e., the identifiers) of the keys given to the corresponding node.

Combinatorial Set Systems (aka Designs)

- A set system is a pair $(X, \mathcal{A})$, where the elements of X are called points and $\mathcal{A}$ is a set of subsets of X, called blocks.
- As stated above, we pair up the blocks of the set system with the nodes in the WSN, and the points in the block are the key identifiers of the keys given to the corresponding node.
- The degree of a point $x \in X$ is the number of blocks containing x
- $(X, \mathcal{A})$ is regular (of degree r) if all points have the same degree, r; then each key occurs in r nodes in the WSN.
- If all blocks have size k, then $(X, \mathcal{A})$ is said to be uniform (of rank k); then each node is assigned k keys.

Toy Example

We list the blocks in a projective plane of order 2 and the keys in the corresponding KPS. This design has seven points, seven blocks, is regular of degree 3 and uniform of rank 3. Further, every pair of points occurs in a unique block and every pair of blocks intersect in a unique point.

node	block	key assignment
$\mathbf{N_1}$	$\{1,2,4\}$	$\mathbf{key}_1, \mathbf{key}_2, \mathbf{key}_4$
$\mathbf{N_2}$	$\{2,3,5\}$	$\mathbf{key}_2, \mathbf{key}_3, \mathbf{key}_5$
$\mathbf{N_3}$	$\{3,4,6\}$	$\mathbf{key}_3, \mathbf{key}_4, \mathbf{key}_6$
$\mathbf{N_4}$	$\{4,5,7\}$	$\mathbf{key}_4, \mathbf{key}_5, \mathbf{key}_7$
$\mathbf{N_5}$	$\{1,5,6\}$	$\mathbf{key}_1, \mathbf{key}_5, \mathbf{key}_6$
$\mathbf{N_6}$	$\{2,6,7\}$	$\mathbf{key}_2, \mathbf{key}_6, \mathbf{key}_7$
$\mathbf{N_7}$	$\{1,3,7\}$	$\mathbf{key}_1, \mathbf{key}_3, \mathbf{key}_7$

The values of keys are secret, but the lists of key identifiers (the blocks) are public. It is easy to see that $\mathbf{Pr_1} = 1$ and $\mathbf{fail_1} = 1/5$.

Possible Advantages of Deterministic KPS

Deterministic KPS have several possible advantages:

Simpler set-up

No random number generator is required for key assignment; simple formulas dictate which keys are given to which nodes.

No need to verify expected properties of the WSN

Randomized KPS have desirable properties with high probability, but there are no guarantees, e.g., due to a "bad" choice of random numbers.

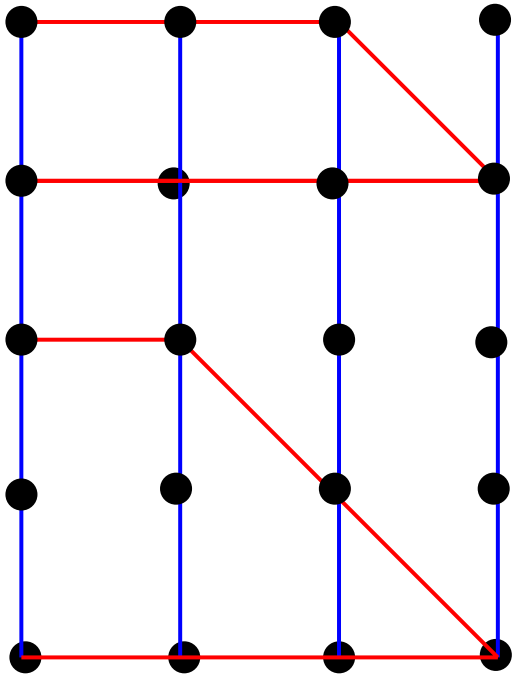
Simpler shared-key discovery and path-key establishment

The complexity of these algorithms can be significantly reduced, sometimes to $O(1)$ time, (as compared to $O(k)$ or $O(k \log k)$ time required in the randomized case).

Transversal Designs

- Lee and Stinson (2005) proposed using transversal designs to construct KPS.
- Let n, k and t be positive integers.
- A transversal design TD(t, k, n) is a triple $(X, \mathcal{H}, \mathcal{A})$, where X is a finite set of cardinality kn, $\mathcal{H}$ is a partition of X into k parts (called groups) of size n, and $\mathcal{A}$ is a set of k-subsets of X (called blocks), which satisfy the following properties:
  1. $|H \cap A| = 1$ for every $H \in \mathcal{H}$ and every $A \in \mathcal{A}$, and
  2. every t elements of X from different groups occurs in exactly one block in $\mathcal{A}$.
- Transversal designs are equivalent to orthogonal arrays, which have been extensively studied in the setting of statistical design of experiments.

Some Blocks in a Transversal Design (Diagram)



Groups are represented as vertical blue lines, and blocks are represented as red lines. Each block is a transversal of the groups.

An Easy Construction for Transversal Designs

- Suppose that p is prime and $t \leq k \leq p$.
- A TD(t, k, p) is constructed by evaluating the p^t **polynomials** of degree at most $t-1$ over $\mathbb{Z}_p$ at k distinct points of $\mathbb{Z}_p$.
- Define
$$X = \{0, \ldots, k-1\} \times \mathbb{Z}_p.$$
- For every $\mathbf{c} = (c_0, \ldots, c_{t-1}) \in (\mathbb{Z}_p)^t$, define a block
$$A_{\mathbf{c}} = \left\{ \left(x, \sum_{i=0}^{t-1} c_i x^i \right) : 0 \leq x \leq k-1 \right\}.$$
- Let
$$\mathcal{A} = \{A_{\mathbf{c}} : \mathbf{c} \in (\mathbb{Z}_p)^t\}.$$
- In the case $t = 2$, the resulting KPS are called **linear KPS**.

Properties of the Linear KPS

- A TD$(2, k, n)$ gives rise to a KPS where

$$\mathbf{Pr_1} = \frac{k}{n+1} \quad \text{and} \quad \mathbf{fail_1} = \frac{n-2}{n^2-2}.$$

- The Eschenauer-Gligor scheme has

$$\mathbf{Pr_1} \approx \frac{k^2}{v} \quad \text{and} \quad \mathbf{fail_1} \approx \frac{k}{v}$$

  when $v \gg k$.

- Since $v = nk$ in a TD$(2, k, n)$, the two schemes have very similar properties.

ID-based One-way Function KPS

- We now consider *ID-based One-way Function KPS*, which were introduced by Lee and Stinson (2005).
- Suppose every pairwise key $L_{u,v}$ is computed as

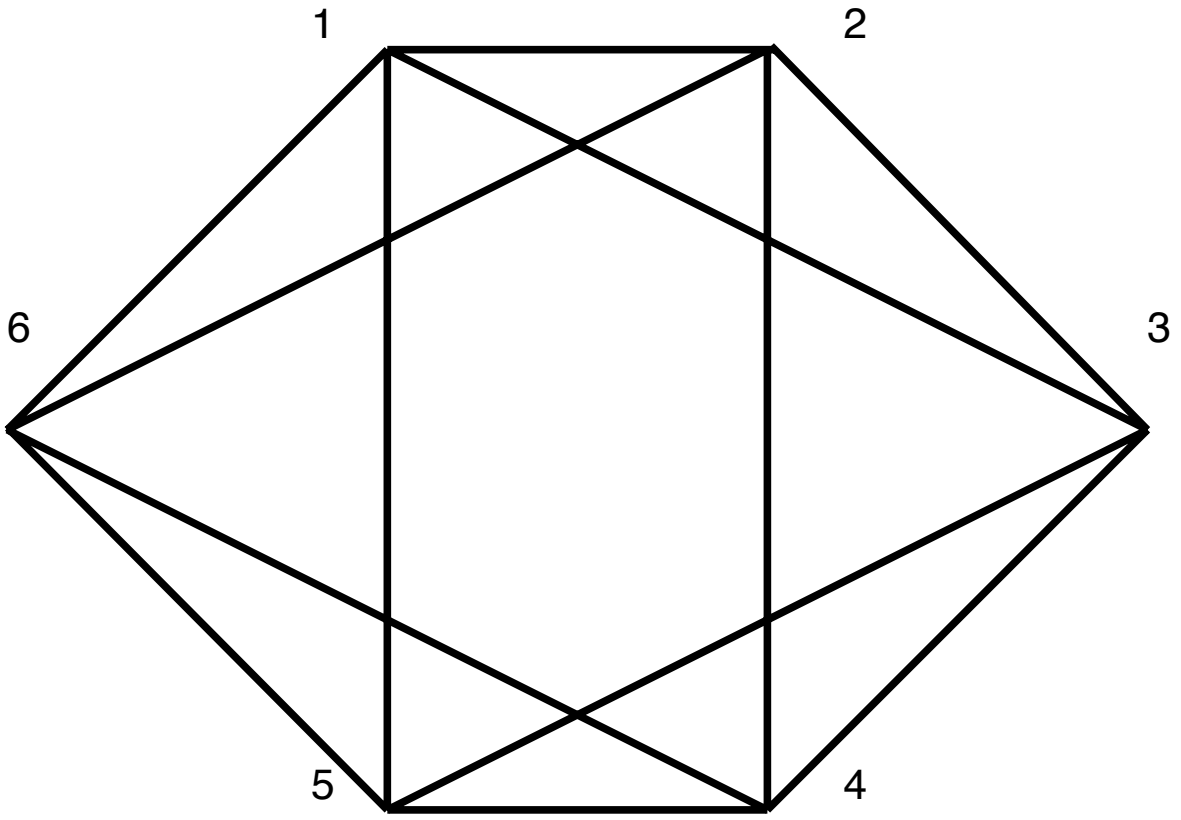
$$L_{u,v} = L_{v,u} = h(R_{\{u,v\}} \parallel \mathrm{ID}(u) \parallel \mathrm{ID}(v)),$$

  where
  - for every $u \in \mathcal{U}$, $\mathrm{ID}(u)$ denotes public identifying information for node u, and
  - $u < v$ (this requirement ensures that $L_{u,v} = L_{v,u}$).
- Each of nodes u and v must be given the value of **either** $R_{\{u,v\}}$ **or** $L_{u,v}$.
- If node u is given $R_{\{u,v\}}$ and node v is given $L_{u,v}$, then the value $R_{\{u,v\}}$ can be "re-used" for some other key $L_{u,w}$, thus reducing the storage requirement of node u.

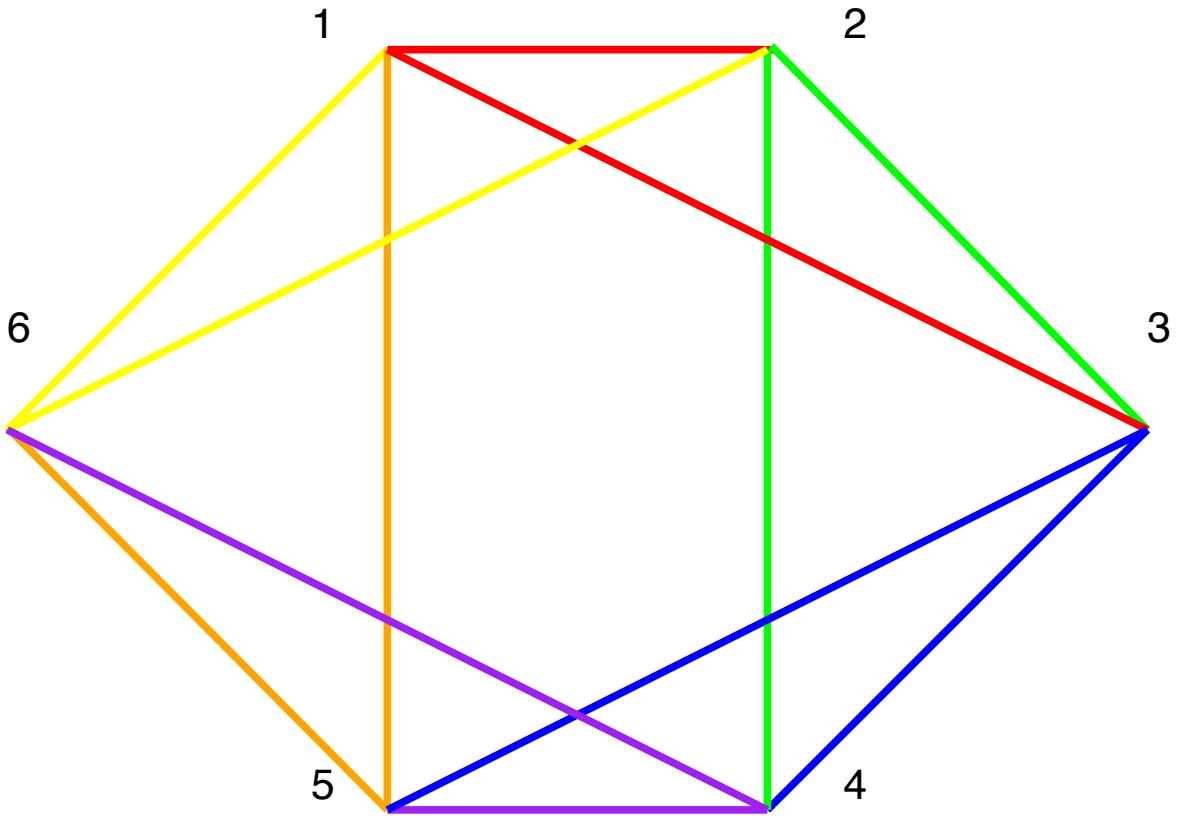
Edge-decompositions into Stars

- Suppose that G is the desired communication graph of the KPS.
- Suppose we find an edge-decomposition of G into **stars** (i.e., complete bipartite graphs $K_{1,m}$ (m is not fixed).
- There is a secret random value associated with the **centre** of each star, which allows the centre to compute m different keys.
- The **leaves** get the actual key values.
- The scheme will be secure against coalitions of **arbitrary size** provided the key derivation function h is "secure".
- Security can be proven formally in the **random oracle model**.

Example: a Communication Graph



Example (cont.): An Edge-decomposition into Six Stars



The Resulting KPS

In the resulting KPS, each node stores **one secret value** (corresponding to the star for which it is the centre) and **two keys** (corresponding to the two stars for which it is a leaf).
The twelve keys are:

$L_{1,2} = h(R_1 \parallel \mathrm{ID}(1) \parallel \mathrm{ID}(2))$	$L_{1,3} = h(R_1 \parallel \mathrm{ID}(1) \parallel \mathrm{ID}(3))$
$L_{2,3} = h(R_2 \parallel \mathrm{ID}(2) \parallel \mathrm{ID}(3))$	$L_{2,4} = h(R_2 \parallel \mathrm{ID}(2) \parallel \mathrm{ID}(4))$
$L_{3,4} = h(R_3 \parallel \mathrm{ID}(3) \parallel \mathrm{ID}(4))$	$L_{3,5} = h(R_3 \parallel \mathrm{ID}(3) \parallel \mathrm{ID}(5))$
$L_{4,5} = h(R_4 \parallel \mathrm{ID}(4) \parallel \mathrm{ID}(5))$	$L_{4,6} = h(R_4 \parallel \mathrm{ID}(4) \parallel \mathrm{ID}(6))$
$L_{5,6} = h(R_5 \parallel \mathrm{ID}(5) \parallel \mathrm{ID}(6))$	$L_{5,1} = h(R_5 \parallel \mathrm{ID}(1) \parallel \mathrm{ID}(5))$
$L_{6,1} = h(R_6 \parallel \mathrm{ID}(1) \parallel \mathrm{ID}(6))$	$L_{6,2} = h(R_6 \parallel \mathrm{ID}(2) \parallel \mathrm{ID}(6))$

The nodes store the following information:

$1 : R_1, L_{5,1}, L_{6,1}$	$2 : R_2, L_{6,2}, L_{1,2}$	$3 : R_3, L_{1,3}, L_{2,3}$
$4 : R_4, L_{2,4}, L_{3,4}$	$5 : R_5, L_{3,5}, L_{4,5}$	$6 : R_6, L_{4,6}, L_{5,6}$

Storage Requirements

- The **storage** $s(u)$ required by a node u is equal to the number of stars that contain the node u.
- The **total storage** is defined to be

$$S = \sum_{u \in V} s(u).$$

Theorem (Paterson and Stinson, 2014)

The optimal total storage for an ID-based One-way Function KPS realizing a communication graph $G = (V, E)$ is

$$S^* = |V| + |E| - \alpha(G),$$

where $\alpha(G)$ denotes the size of a maximum independent set of vertices in G.

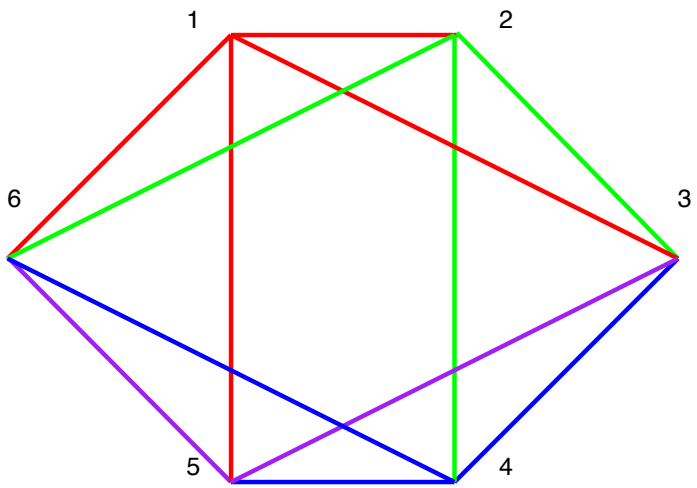
The above theorem shows that computing S^* is NP-hard.

Constructing Schemes with Optimal Storage Requirements

- Suppose we can find an independent set of size $\alpha(G)$, say $W \subseteq V$.
- We use a greedy algorithm to find the star decomposition.
- For every vertex $u \in V \setminus W$, choose a star with centre u using **all the edges incident with v that have not previously been selected**.
- This process will use up all the edges, since every edge is incident with at least one vertex in $V \setminus W$.
- This gives a KPS with total storage equal to $|V| + |E| - \alpha(G)$.

Example

- In the example graph, $|V| = 6$, $|E| = 12$ and $\alpha(G) = 2$, so the optimal total storage is $6 + 12 - 2 = 16$.
- The previous decomposition yielded total storage equal to 18.
- $\{3, 6\}$ is an independent set of size 2.
- We greedily construct stars with centres $1, 2, 4$ and 5:
  - $12, 13, 15, 16$
  - $23, 24, 26$
  - $43, 45, 46$
  - $53, 56$



Maximum Storage

- The **maximum storage** is defined to be

$$S_{\max} = \max\{s(u) : u \in V\}.$$

Theorem (Paterson and Stinson, 2014)

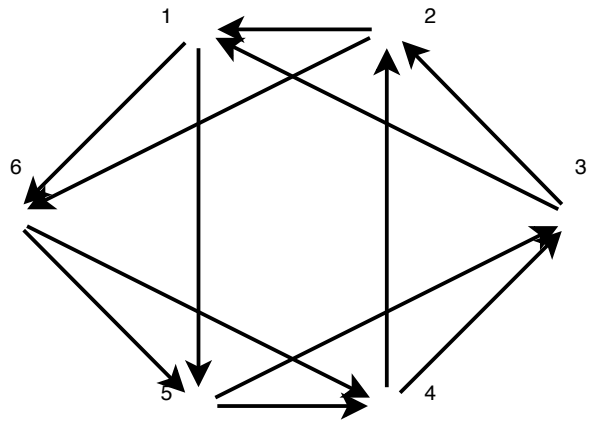
The optimal maximum storage for a scheme realizing a d-regular communication graph $G = (V, E)$ is

$$S^*_{\max} = \left\lfloor \frac{d+3}{2} \right\rfloor.$$

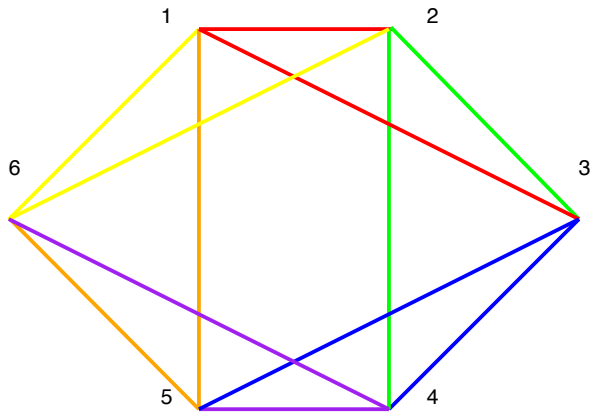
Eulerian Circuits and Optimal Maximum Storage for Regular Graphs

- Lee and Stinson (2005) used **Eulerian circuits** to construct KPS with optimal maximum storage in the case of regular graphs of even degree.
- The upper bound on maximum storage is $S^*_{\max} \leq (d+2)/2$ when d is even.
- Orient the edges in E so a (directed) Eulerian circuit is obtained.
- For every vertex u, there are $d/2$ edges directed into u and $d/2$ edges directed away from u.
- The edges directed **into** vertex u form a star $K_{1,d/2}$.
- In the resulting star decomposition, every vertex is in $d/2+1$ stars.

Example



In the example graph, it is easily seen that $216531542643 2$ is an Eulerian circuit that gives rise to the first star decomposition. The resulting KPS has optimal maximum storage equal to 3.



Optimal Maximum Storage for General Graphs

- Paterson and Stinson (2014) have shown how the optimal maximum storage can be computed in polynomial time for arbitrary communication graphs G.
- Computing optimal maximum storage is closely related to the well-studied **minimum maximum outdegree** problem.
- This problem requires directing the edges of G to minimize the maximum outdegree of a vertex of the resulting directed graph, which is denoted by $\mathrm{MMO}(G)$.
- It is easy to see that

$$\mathrm{MMO}(G) \leq S^*_{\max}(G) \leq \mathrm{MMO}(G) + 1.$$

- So there are **two possible values** for $S^*_{\max}(G)$, and it turns out that we can determine the correct value in polynomial time!

References

[1] Rolf Blom. Non-public key distribution. In *Proceedings of CRYPTO '82*, pages 231–236, Plenum Press, 1983.

[2] Jooyoung Lee and Douglas R. Stinson. Deterministic key predistribution schemes for distributed sensor networks. *Lecture Notes in Computer Science* **3357** (2005), 294–307 (SAC 2004 Proceedings).

[3] Jooyoung Lee and Douglas R. Stinson. A combinatorial approach to key predistribution for distributed sensor networks. *IEEE Wireless Communications and Networking Conference (WCNC 2005)*, vol. 2, pp. 1200–1205.

[4] Maura B. Paterson and Douglas R. Stinson. Optimal constructions for ID-based one-way-function key predistribution schemes realizing specified communication graphs. Preprint, 2014.

Thank You For Your Attention!