The Combinatorics of Cryptographic Key Establishment: a short tour

Keith Martin

Information Security Group, Royal Holloway, University of London

21st British Combinatorial Conference, July 2007

Confessions

I used to be a pure mathematician but now I've...

- Written a paper in Microsoft Word
- Worked in an Electrical Engineering Department
- Taught a course where the mathematical highlight is the Euclidean Algorithm
- Published a paper called *Evaluation of authentication protocols for mobile environment value-added services*
- Found myself sitting in mathematics seminars thinking *but is this even vaguely useful?*

The plan

1. Cryptography and key management
2. Framework for key establishment
3. Key predistribution schemes
4. Key distribution schemes
5. Key agreement schemes

A few caveats before we start

- This is not a complete survey of key establishment (how long have I got?)
- The primary focus is on key establishment areas that have been influenced by combinatorial mathematics.
- As a result we will largely (but not exclusively) be dealing with symmetric cryptography.
- All the schemes mentioned are of theoretical interest (but don't implement them at home before checking their applicability!)

Cryptography and key management

(all you need to know, and hopefully no more than that...)

Cryptography

The science of **cryptography** provides a toolkit of mathematical techniques, algorithms and protocols to provide core electronic security services such as:

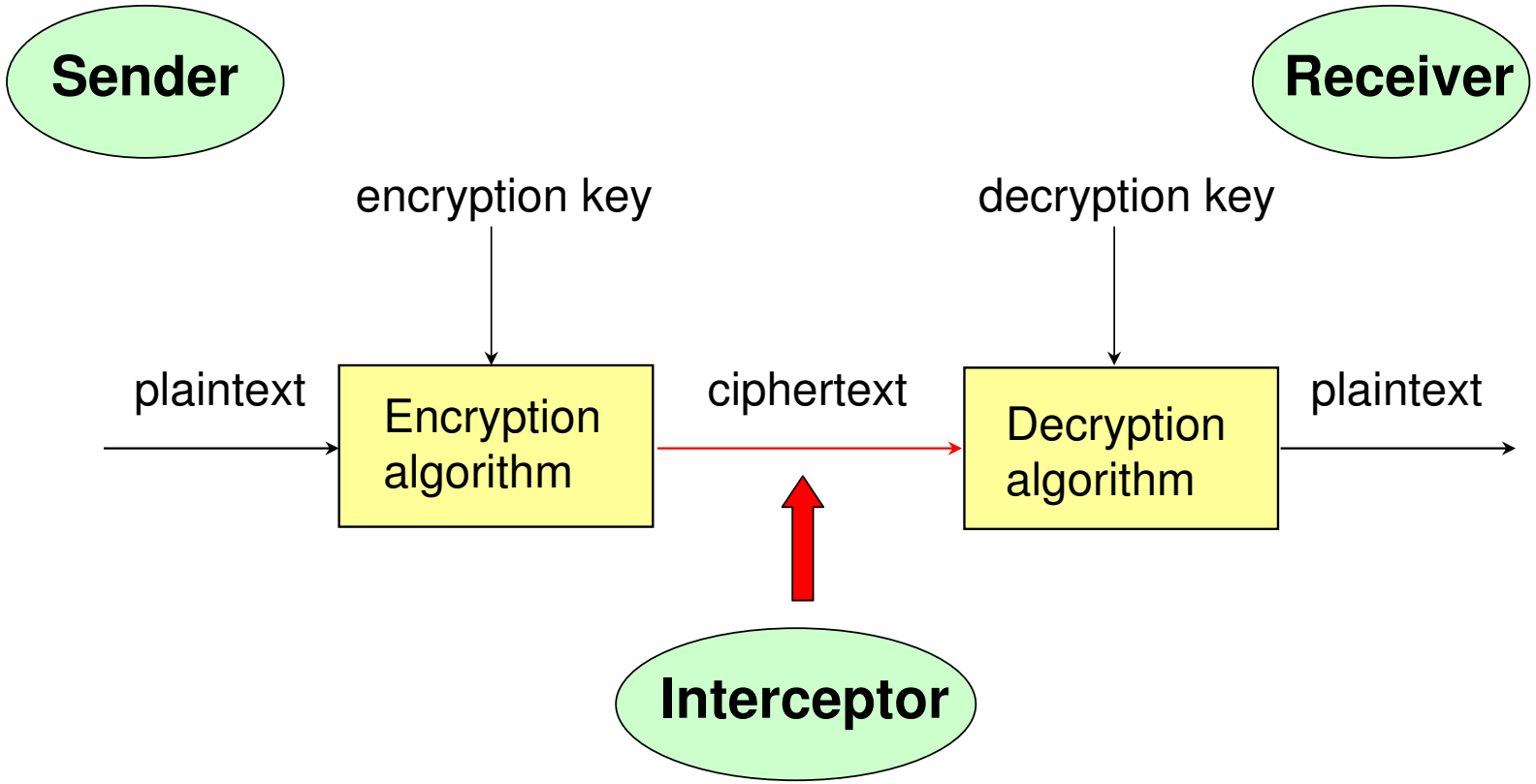
- confidentiality
- data integrity
- data origin (message) authentication
- non-repudiation

Cryptographic applications

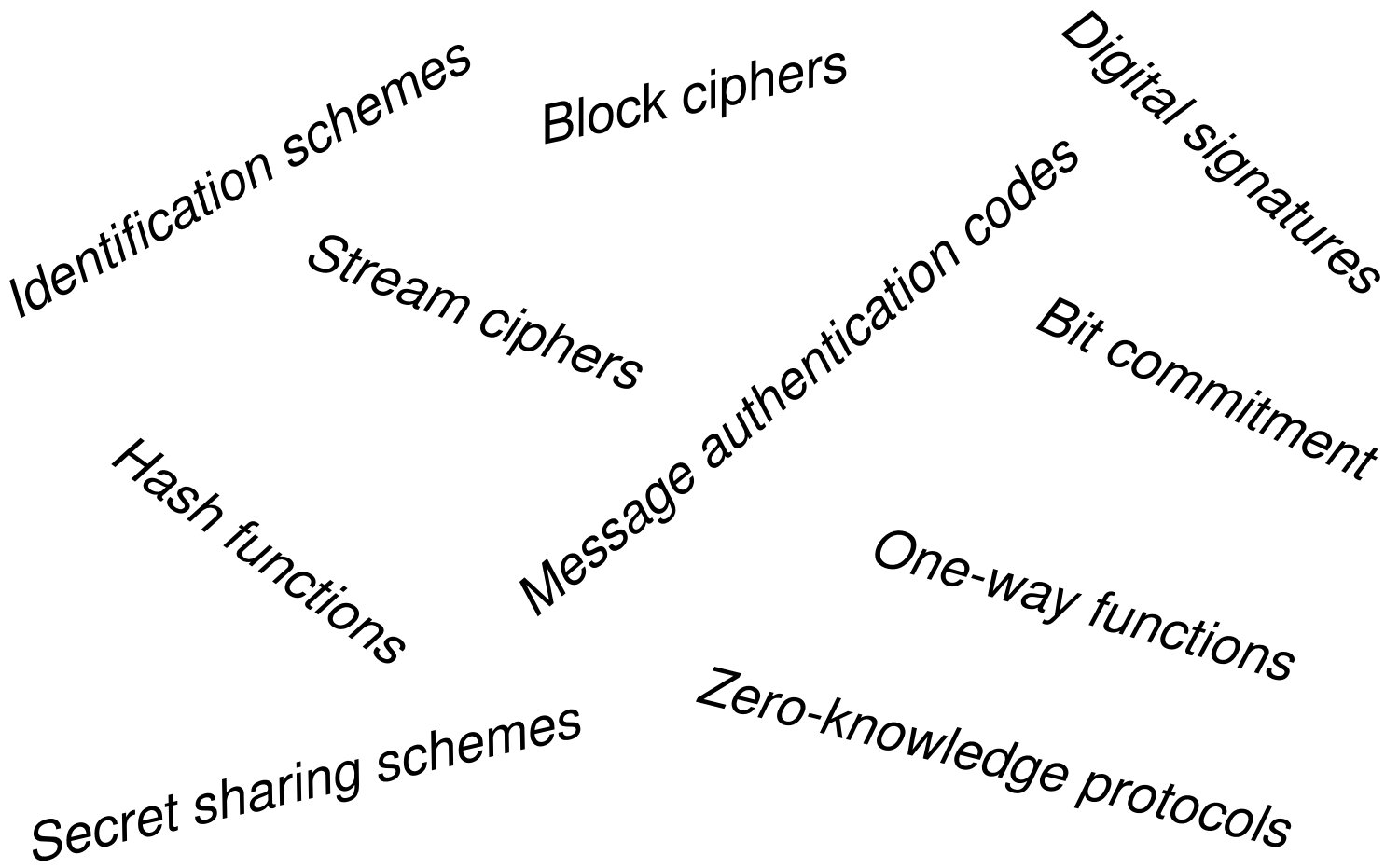
Deployment of cryptography is increasingly ubiquitous:

- ATM security
- mobile communications security
- Internet transactions
- smart card applications
- hard disk protection
- ...

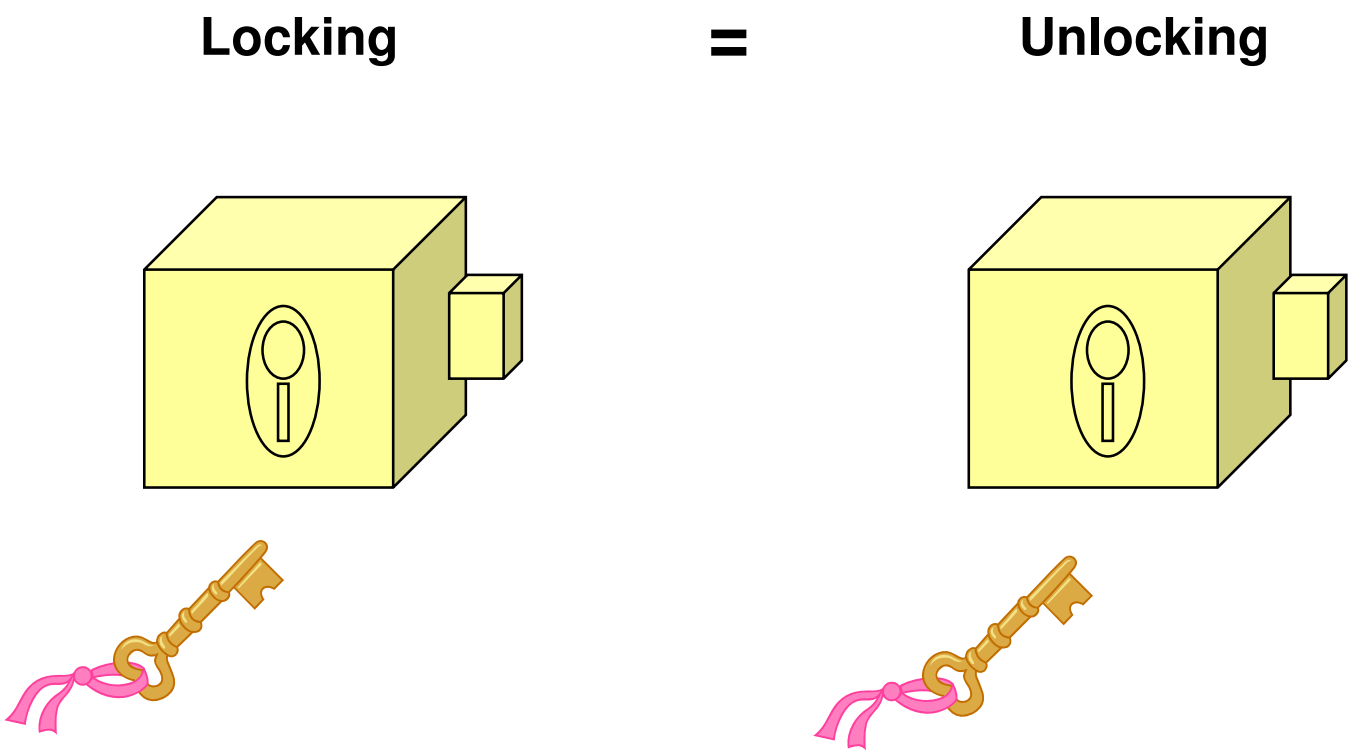
Basic model of a cryptosystem



Cryptographic primitives

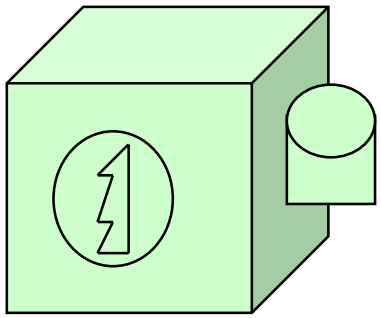


Symmetric cryptography

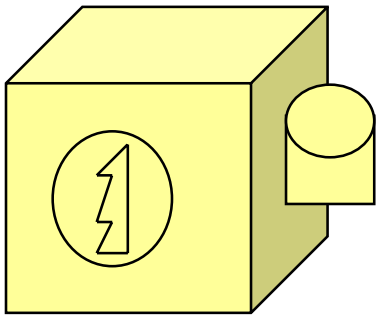


Public key cryptography

Anyone can lock

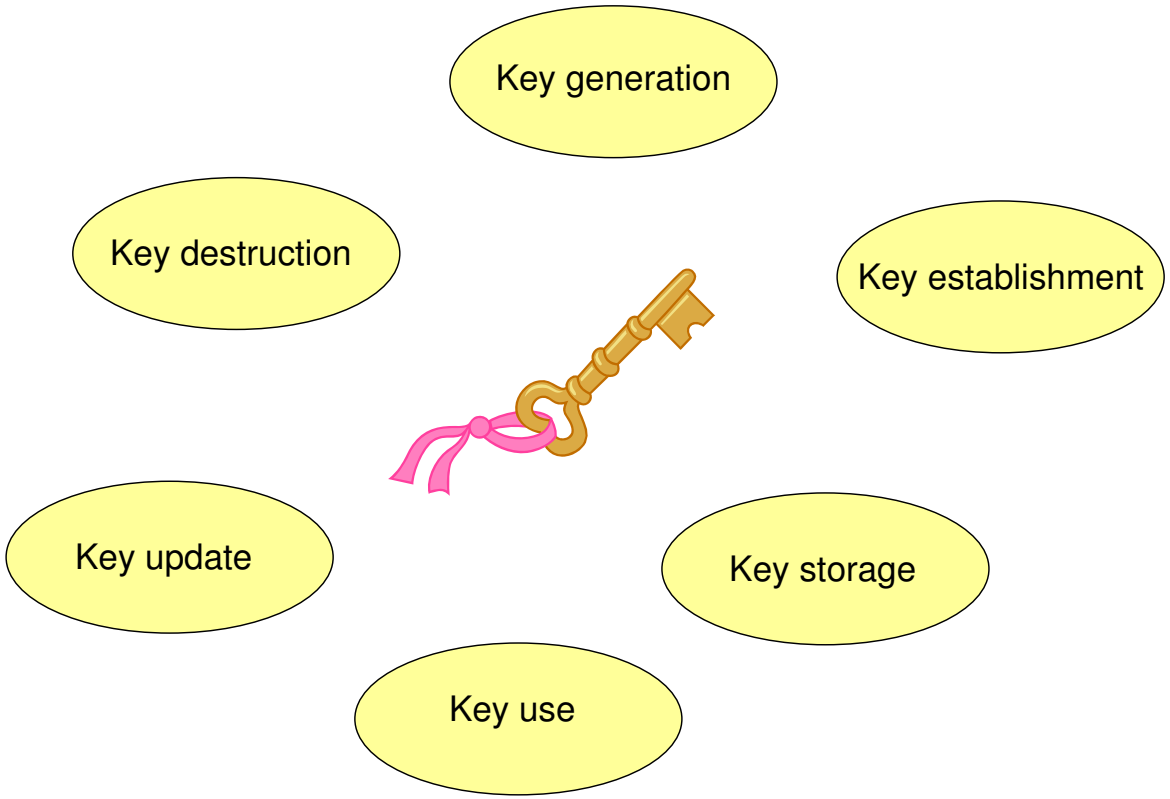


Only a key holder can unlock



Key management

Key management is a generic term that is associated with the entire lifecycle of cryptographic keys:



Framework for key establishment

(Lots of definitions and no mathematics...)

Fundamental definitions

- **Users** $\mathcal{U}$: the set of entities in a network.
- **Communication structure** $\mathcal{C}$: collection of subsets of $\mathcal{U}$ for whom we wish to establish cryptographic keys.
- **Group key** k_A: a cryptographic key shared by a set A of users belonging to $\mathcal{C}$.
- **Trusted Authority** (TA): an entity regarded as trustworthy and secure by all users in the network.

Key establishment

A **key establishment scheme** for communication structure $\mathcal{C}$ is a set of protocols that allow any set $A \in \mathcal{C}$ to establish a group key k_A. It consists of the following operational phases:

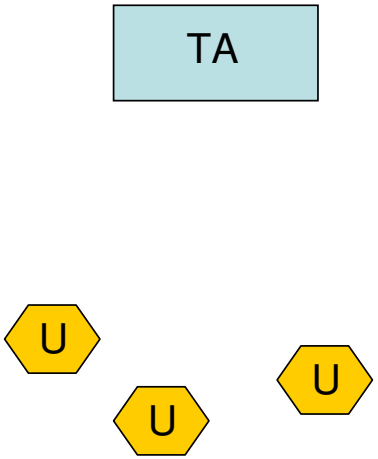
1. **Initialisation**: (A trusted authority) generates all the data required to initialise the scheme, including
   - secret data u_i specific to each user U_i
   - public system-wide data Pub.
2. **Key establishment**: Users $A \in \mathcal{C}$ establish their common key k_A.
3. **Update**: In this optional phase, the secret and public data are modified, either because:
   - the communication structure has changed.
   - the original keys have expired.

Classification

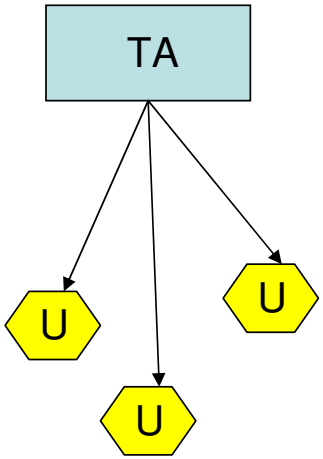
A major distinguisher between key establishment schemes is the extent to which communication between entities occurs during the key establishment phase. Three operational environments are:

1. *Isolation*: Users have no communication channels available to support key establishment and thus must be able to do so on their own. We refer to such schemes as **group key predistribution schemes**.
2. *TA-user*: The TA has some ability to communicate with users during the key establishment phase. We refer to such schemes as **group key distribution schemes**.
3. *User-user*: Users have some ability to communicate with one another during the key establishment phase. We refer to such schemes as **group key agreement schemes**.

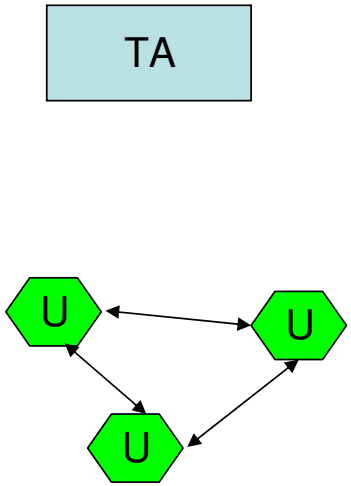
Classification



Key Predistribution



Key Distribution



Key Agreement

Secondary distinguishers

Key establishment schemes differ with respect to:

- security model
- deterministic v probabilistic
- properties of group keys
- extended capabilities

Security model

The main threat to security of a key establishment scheme is the ability of users (or outside parties) to obtain a group key that they are not entitled to.

The security model can be classified by:

1. *Type of security*: Most commonly either **unconditional** or **computational** security.
2. *Resilience*: The degree of resilience to collusion between other users in the scheme. Common examples of **exclusion structures** are:
   - **Full collusion security**: $\mathcal{X}$ consists of all subsets of $\mathcal{U}$.
   - w**-security**: $\mathcal{X}$ consists of all subsets of $\mathcal{U}$ of at most size w.

Deterministic v probabilistic

Key establishment schemes are either:

- **Deterministic**: we can guarantee that a group $A \in \mathcal{C}$ is able to establish a common key.
- **Probabilistic**: we can only guarantee that a group $A \in \mathcal{C}$ is able to establish a common key with a certain probability.

Properties of group keys

Group keys can be:

- **Predistributed**: if k_A is a function only of the secret values $\{u_i \,|\, U_i \in A\}$ and Pub.
- **Prepositioned**: if k_A is predetermined by the values $\{u_i \,|\, U_i \in A\}$ and Pub, but that its actual value cannot be computed until some further processing occurs during the key establishment phase.
- **Independent**: if knowledge of other group keys provides no information about the value of k_A.
- **Combinatorial**: if k_A can be represented as a subset of the collective secret user information of users belonging to A.

Extended capabilities

Application environments vary with respect to their need for:

- **Flexibility**: the extent to which a key establishment scheme is able to efficiently accommodate an update phase.
- **Computational capability**: the extent to which entities (particularly users) have the ability to perform computations.
- **Decentralisation**: whether roles normally conducted by the TA are required to be distributed amongst a number of separate entities. This can be for reasons of scalability, security or reliability.
- **Collaboration**: the degree of collaboration that is required (or permitted) to take place between users order to establish a group key.

Extended capabilities ctd.

- **Robustness**: a stronger security model might be required for applications where either the TA or users are not trusted to perform their operations honestly.
- **Temporal restrictions**: whether key establishment for certain groups is restricted to specific time intervals or limited to a finite number of key establishment events.
- **Traceability**: whether it is possible to identify fraudulent users who abuse the key establishment scheme.

Evaluation criteria

Functionally similar group key establishment schemes can be compared by evaluating:

- **Secret storage**: the amount of information that a user needs to keep secure.
- **Public storage**: the amount of public information that needs to maintained to operate the scheme.
- **Communication costs**: the amount of data that needs to be exchanged (whether by expensive secure channels or less expensive broadcast channels).
- **Computational costs**: the computational requirements for users in the scheme.

These are particularly important for applications where users are represented by small mobile devices.

Key predistribution

(doing all the hard work first...)

Key predistribution schemes

A $(\mathcal{C}, \mathcal{X})$**-key predistribution scheme** (KPS) is a key establishment scheme with communication structure $\mathcal{C}$ and exclusion structure $\mathcal{X}$ such that:

1. Given $A \in \mathcal{C}$, any $U_i \in A$ can compute the group key k_A from knowledge of u_i and Pub.
2. Given disjoint sets $B \in \mathcal{X}$ and $A \in \mathcal{C}$, it is not possible to compute the group key k_A from knowledge of u_B and Pub (where $u_B = \{u_i \,|\, U_i \in B\}$).

If $\mathcal{C} = \{A \subseteq \mathcal{U} \,|\, |A| = t\}$ and $\mathcal{X} = \{A \subseteq \mathcal{U} \,|\, |A| \leq w\}$ then we will also refer to a $(\mathcal{C}, \mathcal{X})$-KPS as a (t, w)-KPS.

Key predistribution

- Fundamental schemes
- BDVHKY schemes
- Key distribution patterns

Fundamental KPS (1)

A **trivial key predistribution scheme** (TKPS) has the following properties:

- $u_i = \{k_A \,|\, U_i \in A, A \in \mathcal{C}\}$;
- $Pub = \emptyset$;
- $k_A \in u_i$ if and only if $U_i \in A$.

Properties: deterministic, has independent keys, unconditionally secure, full collusion security, can be established for an arbitrary communication structure $\mathcal{C}$.

Features: large user storage, key refreshment requires re-initialisation, minimal public data.

Fundamental KPS (2)

A **trivial key encrypting key predistribution scheme** (TKEKPS) has the following properties:

- $u_i = \{K_A \,|\, U_i \in A, A \in \mathcal{C}\}$, where each K_A is randomly chosen from $\{0,1\}^l$;
- $Pub = \{E_{K_A}(k_A) \,|\, A \in \mathcal{C}\}$;
- $K_A \in u_i$ if and only if $U_i \in A$, with k_A obtained by decrypting $E_{K_A}(k_A)$.

Properties: deterministic, has independent keys, computationally secure, full collusion security, can be established for an arbitrary communication structure $\mathcal{C}$.

Features: large user storage, key refreshment can be conducted by updating public information.

Fundamental KPS (3)

A **direct key encrypting key predistribution scheme** (DKEKPS) has the following properties:

- $u_i = k_i$, where each k_i is randomly chosen from $\{0,1\}^l$;
- $Pub = \{E_{k_i}(k_A) \,|\, U_i \in A, A \in \mathcal{C}\}$;
- $E_{k_i}(k_A) \in Pub$ if and only if $U_i \in A$, with k_A obtained by decrypting $E_{k_i}(k_A)$.

Properties: deterministic, has independent keys, computationally secure, full collusion security, can be established for an arbitrary communication structure $\mathcal{C}$.

Features: large public data, minimal user storage.

Fundamental KPS(4)

A **node-based key predistribution scheme** (NBKPS) has the following properties:

- $Pub = \cup_{1 \leq i \leq n} Pub_i$, where Pub_i is associated with user U_i;
- $u_i = f(Pub_i)$ for some secret function f known only to the TA, which is chosen in such a way that there exists a public function g such that for any $A \in \mathcal{C}$ and any pair $U_i, U_j \in A$ we have that $g(u_i, Pub_A) = g(u_j, Pub_A) = k_A$ (where $Pub_A = \cup_{U_i \in A} Pub_i$).
- By choice of f and g it follows that any $U_i \in A$ can compute k_A.

Properties: depends on instantiation.

Fundamental KPS (5)

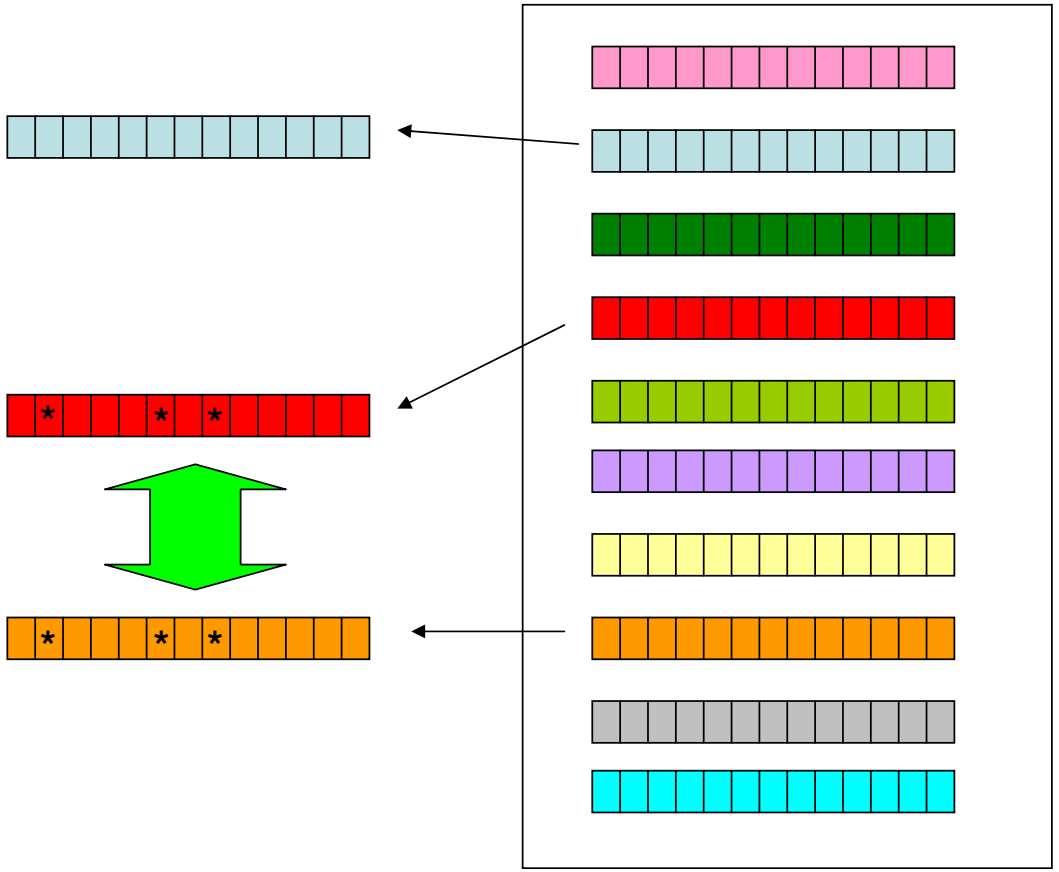
Let $\mathcal{I} = \{x_i \,|\, 1 \leq i \leq v\}$ be a set of v identifiers, each of which is associated by means of a secret function f with a randomly chosen key $k_i = f(x_i)$ from a set $\mathcal{K}$. Let $\mathcal{B}$ be a collection of subsets of $\mathcal{I}$.

We refer to $\mathcal{R} = (\mathcal{I}, \mathcal{B})$ as a **key ring**.

A **key ring predistribution scheme** (KRPS) based on key ring $\mathcal{R} = (\mathcal{I}, \mathcal{B})$ is an NBKPS such that:

- $Pub_i = B_i$ is randomly chosen from $\mathcal{B}$ (such that $B_i \neq B_j$ if $i \neq j$);
- $u_i = \{k_j \,|\, x_j \in B_i\}$;
- $\mathcal{C} \subseteq \{A \subseteq \mathcal{U} \,|\, \cap_{U_i \in A}\, u_i \neq \emptyset\}$;
- For $A \in \mathcal{C}$, group key $k_A = g(\cap_{U_i \in A} k_i)$ for some public combining function g.

Key ring predistribution scheme



BDVHKY (Blundo et al, Crypto 92)

- $Pub_i = s_i$, where $s_i \in GF(q)$ $(q \geq n)$ and $Pub_i \neq Pub_j$ if $i \neq j$;
- The TA (randomly) constructs a secret t-variate polynomial f with coefficients from $GF(q)$,

$$f(x_1, \ldots, x_t) = \sum_{i_1=0}^{w} \cdots \sum_{i_t=0}^{w} a_{i_1 \ldots i_t} x_1^{i_1} \ldots x_t^{i_t},$$

  where $a_{i_1 \ldots i_t} = a_{j_1 \ldots j_t}$ for any permutation $(j_1 \ldots j_t)$ of the indices $\{i_1, \ldots, i_t\}$.
- $u_i = f(Pub_i, x_2, \ldots, x_t) = f(s_i, x_2, \ldots, x_t)$;
- For any $A = \{U_{z_1}, \ldots, U_{z_t}\} \in \mathcal{C}$, the user U_{z_i} computes

$$k_A = u_{z_i}(s_{z_1}, \ldots, s_{z_{i-1}}, s_{z_{i+1}}, \ldots, s_{z_t}) = f(s_{z_1}, \ldots, s_{z_t}).$$

BDVHKY example for $t = 2$ and $w = 1$

- Let $Pub_1 = 1$ and $Pub_2 = 2$
- The TA secretly selects $f(x_1, x_2) = 1 + 3x_1 + 3x_2$
- User U_1 receives $f(1, x_2) = 4 + 3x_2$
- User U_2 receives $f(2, x_2) = 7 + 3x_2$
- When U_1 and U_2 wish to establish a common key:
  - U_1 computes $f(1, 2) = 10$
  - U_2 computes $f(2, 1) = 10$

Properties of BDVHKY

- BDVHKY is a deterministic NBKPS (but not a KRPS)
- BDVHKY offers unconditional w-security
- Each user needs to securely store the equivalent of $\binom{t+w-1}{t-1}$ elements of $GF(q)$
- This is the optimally lowest secure storage for any unconditionally secure (t, w)-KPS
- Several variants exist

Key distribution patterns (Mitchell and Piper 87)

A **set system** $(\mathcal{I}, \mathcal{B})$ consists of a set $\mathcal{I}$ of v elements (**points**) and a collection $\mathcal{B}$ of subsets (**blocks**) of $\mathcal{I}$.

Let $(\mathcal{C}, \mathcal{X})$ be a communication and exclusion structure defined on n users. A $(\mathcal{C}, \mathcal{X})$**-key distribution pattern** (KDP) is a set system $(\mathcal{I}, \mathcal{B})$ with $|\mathcal{B}| = n$ (and every user U_i is associated with a block B_i) such that for any disjoint pair $A \in \mathcal{C}$ and $B \in \mathcal{X}$ we have:

$$\bigcap_{U_i \in A} B_i \not\subseteq \bigcup_{U_j \in B} B_j.$$

A $(\mathcal{C}, \mathcal{X})$**-key distribution pattern predistribution scheme** (KDPPS) is a $(\mathcal{C}, \mathcal{X})$-KRPS that arises by employing a $(\mathcal{C}, \mathcal{X})$-KDP as a key ring.

$(2,1)$-KDP example for seven users

Let $\mathcal{I} = \{0,1,2,3,4,5,6\}$ and $\mathcal{B} =$
$\{0,1,2,4\},\{1,2,3,5\},\{2,3,4,6\},\{3,4,5,0\},\{4,5,6,1\},\{5,6,0,2\},\{6,0,1,3\}$.

- $Pub_1 = \{0,1,2,4\}$, $Pub_2 = \{1,2,3,5\}$, etc
- $u_1 = \{k_0, k_1, k_2, k_4\}$, $u_2 = \{k_1, k_2, k_3, k_5\}$, etc
- U_1 and U_2 share keys k_1 and k_2 (and no other user holds both these keys)
- Group key $k_{\{U_1,U_2\}} = g(k_1, k_2)$ for some public combining function g.

Cover free families

A (t, w, d)-**cover-free family** (CFF) is a set system $(\mathcal{I}, \mathcal{B})$ such that for any disjoint sets of t blocks A and w blocks B we have:

$$|\bigcap_{B_i \in A} B_i \setminus \bigcup_{B_j \in B} B_j| \geq d.$$

(t, w, d)-cover-free families give rise to (t, w)-key distribution patterns (Stinson and Wei 04).

(t, w)-KDP research

- Constructions from combinatorial designs (Stinson 97).
- Constructions from finite geometry (O'Keefe 95, Rinaldi 04).
- Constructions from orthogonal and perpendicular arrays (Stinson and Van Trung 98).
- Existence results for probabilistic KDPs (Dyer et al 95).
- Bounds on information storage (Quinn 99).
- Efficiency improvements using resilient functions (Stinson 97).
- Generalised for application to hash trees (Lee and Stinson, SAC05).

A future for KDPs?

KDPs have two interesting properties:

- The shared group keys are combinatorial.
- The rich combinatorial structure lends itself to constructions for more complex communication structures.

Key distribution

(the default way to establish keys...)

Key distribution schemes

A $(\mathcal{C}, \mathcal{X})$**-key distribution scheme** (KDS) is a key establishment scheme with communication structure $\mathcal{C}$ and exclusion structure $\mathcal{X}$ such that:

1. Given $A \in \mathcal{C}$, any $U_i \in A$ can compute the group key k_A from knowledge of u_i and $v_{i,A}$, where $v_{i,A}$ is some information obtained by U_i from the TA during the key establishment phase for key k_A.
2. Given disjoint sets $B \in \mathcal{X}$ and $A \in \mathcal{C}$, it is not possible to compute the group key k_A from knowledge of u_B and v_B (where $u_B = \{u_i \,|\, U_i \in B\}$ and $v_B = \{v_{i,A} \,|\, U_i \in B\}$).

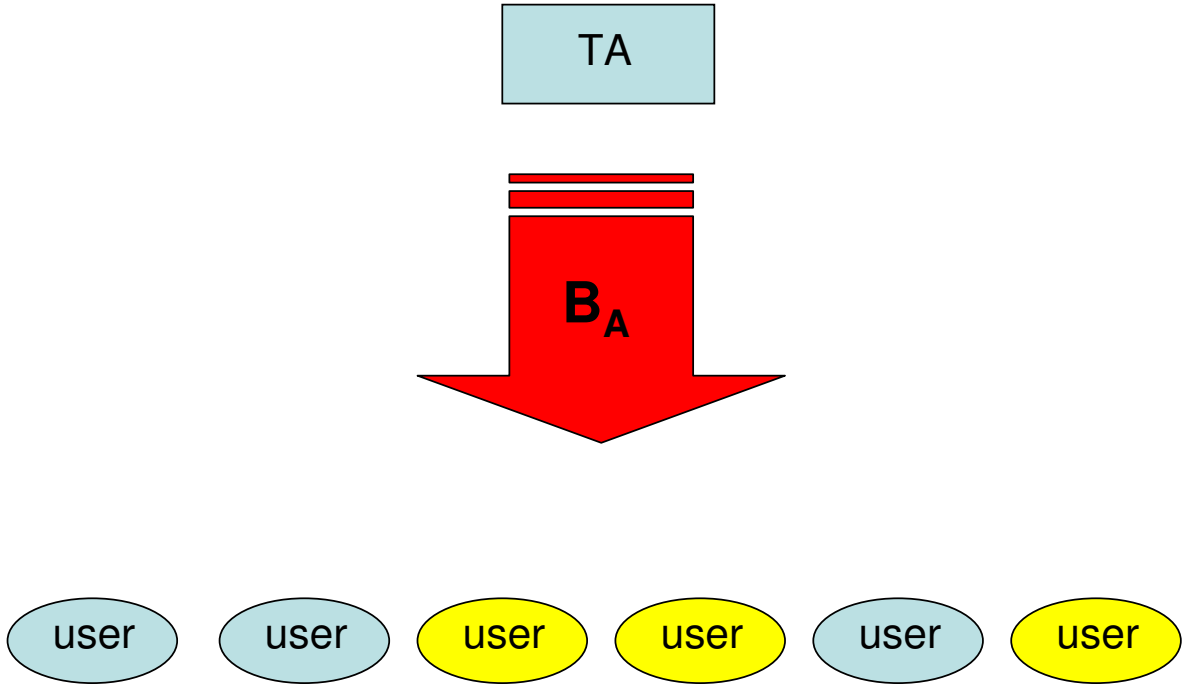
Particularly suited to applications where an online TA exists and group keys k_A are necessarily generated at the time of request.

Broadcast encryption

A $(\mathcal{C}, \mathcal{X})$-**broadcast encryption scheme** (BES) is a key distribution scheme with communication structure $\mathcal{C}$ and exclusion structure $\mathcal{X}$ such that for every user $U_i \in \mathcal{U}$, $v_{i,A} = B_A$, where B_A is a public message broadcast to all users in $\mathcal{U}$ at the start of the key establishment phase for k_A.

- Broadcast encryption schemes were first proposed (Berkovitz, Eurocrypt 91, Fiat and Naor, Crypto 93) with applications such as access to streamed multimedia services in mind.
- Content is encrypted using k_A (where A is the group of users permitted to access the service).
- B_A is broadcast as a header that allows an authorised user U_i in A to determine k_A and hence decrypt the service.

Broadcast encryption scheme



Two flavours of broadcast encryption

1. **General broadcast encryption**:
   - Designed for as large a communication structure as possible, since this maximises the possible groups for whom group keys can be generated.
   - Suitable for *pay-per-view* services, where the groups of users receiving content are highly variable.
2. **Long term group management**:
   - Characterised by a single large group of users that may change gradually over time.
   - Suitable for *subscription* services, where we essentially only ever want to broadcast to the entire group of subscribed users.

Computational capabilities of users

- **Stateless receivers** if the users cannot retain information from previous broadcasts (or have ability to write to memory). This might be the case for example if the user is a set-top decoder.
- **Stateful receivers** if the users can retain information from previous broadcasts (or have the ability to write to memory). In this case if new keys are broadcast then users can replace the keys that were distributed to them on initialisation.

Broadcast encryption from a KPS

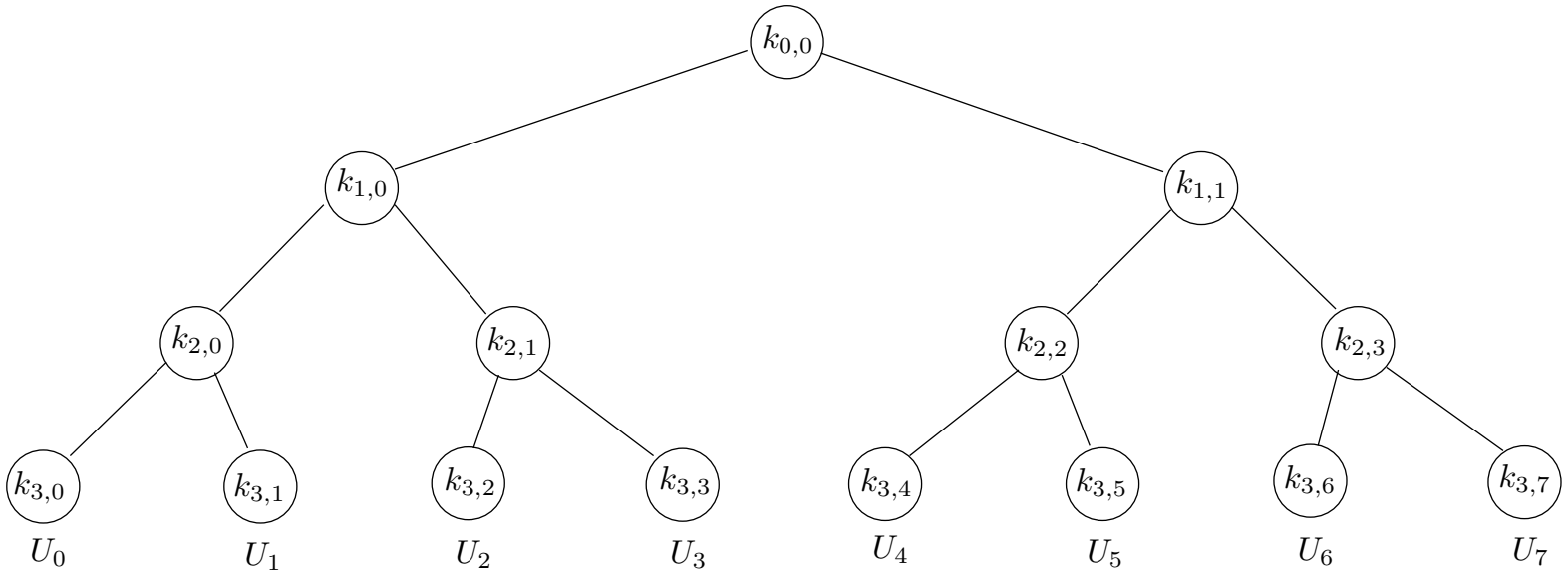
If we have a $(\mathcal{C}, \mathcal{X})$-KPS then we can realise a $(\mathcal{C}, \mathcal{X})$-BES as follows:

- u_i is the same for both the KPS and the BES;
- $B_A = E_{k_A^*}(k_A)$, where k_A^* is the group key for $A \in \mathcal{C}$ in the KPS and k_A is a freshly generated group key for A in the BES;
- Only a user U_i in A can establish k_A^* from u_i and hence decrypt the new group key k_A.

In a series of papers (Stinson 97, Stinson and Van Trung 98, Stinson and Wei 99) KPSs are combined with ideal secret sharing schemes to construct more efficient BESs.

Logical key hierarchies (Wallner et al 98)

This scheme operates on stateful receivers.



For each user U_j let $u_j = \{k_{x,y} \,|\, k_{x,y}$ is on the path from $k_{h,j}$ to $k_{0,0}\}$.

Cover-based revocation systems

Let $(\mathcal{I}, \mathcal{B})$ be a set system and for each $x \in \mathcal{I}$ let $\beta(x) = \{B \in \mathcal{B} \,|\, x \in B\}$. We say that $(\mathcal{I}, \mathcal{B})$ is a **cover-based revocation system** (CBRS) if for every non-empty $\mathcal{A} \subseteq \mathcal{B}$ there exists $\mathcal{I}_{\mathcal{A}} \subseteq \mathcal{I}$ such that

$$\bigcup_{x \in \mathcal{I}_{\mathcal{A}}} \beta(x) = \mathcal{A}.$$

In other words, a set system is a CBRS if for every non-empty collection $\mathcal{A}$ of blocks there exists a subset $\mathcal{H}$ of points such that the subsets $\{\beta(x) \,|\, x \in \mathcal{H}\}$ form a cover of $\mathcal{A}$.

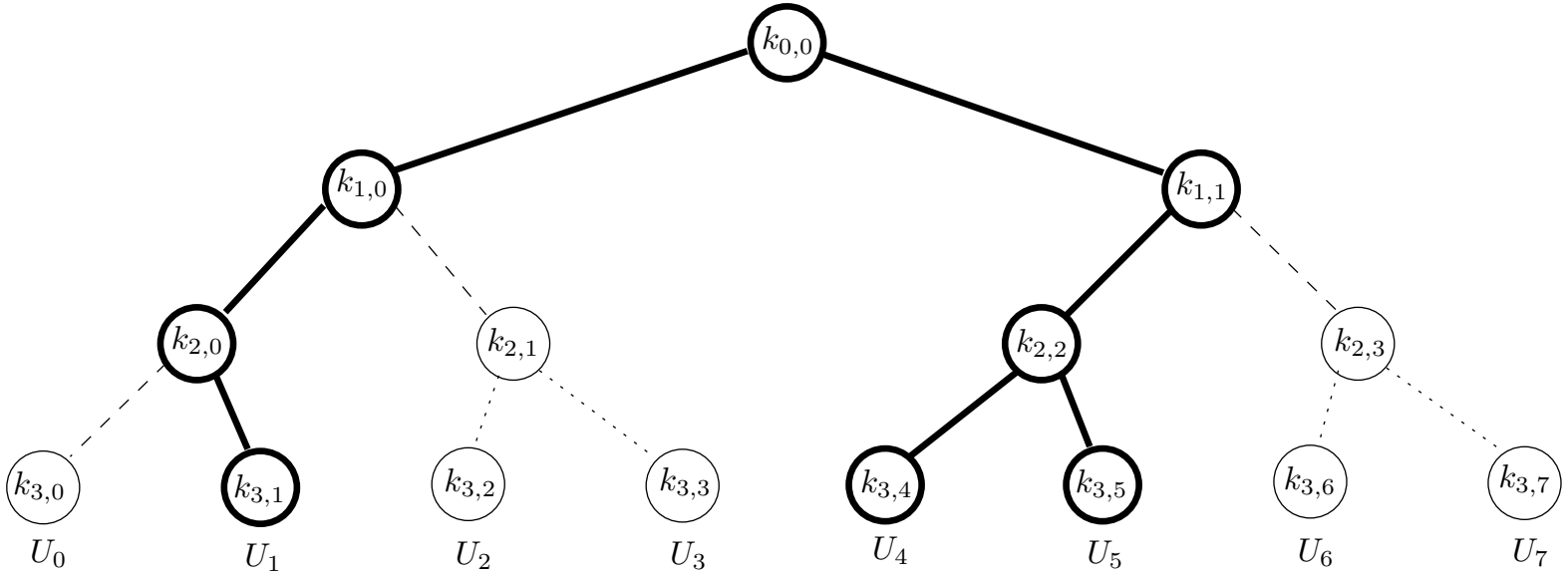
KDSs based on covers for stateless receivers

Given a cover-based revocation system $(\mathcal{I}, \mathcal{B})$ we can define a broadcast encryption scheme for stateless receivers as follows:

- Associate each point $x \in \mathcal{I}$ with a key k_x, and associate each block $B_i \in \mathcal{B}$ with a user U_i.
- $u_i = \{k_x \,|\, x \in B_i\}$.
- For any subset A of users (corresponding to the set of blocks $\mathcal{A}$), $B_A = \{E_{k_x}(k_A) \,|\, x \in \mathcal{I}_{\mathcal{A}}\}$.
- By definition of a CBRS, the only users holding at least one of the keys k_x are those in A.

Such schemes are only practical if there also exists an efficient algorithm for determining the appropriate cover of keys, given a particular subset of users.

Complete subtree method (Naor et al, Crypto 01)



In this case $B_A = \{E_{k_{3,0}}(k_A), E_{k_{2,1}}(k_A), E_{k_{2,3}}(k_A)\}$.

There are many variants of this scheme, for example to a-ary trees (Asano 02).

Snapshot of broadcast encryption research

- *Alternative schemes*: there have been many proposals for trading off storage against broadcast size:
  - **Subset difference method** (Naor et al, Crypto 01).
  - **Layered subset difference method** (Halevy and Shamir 02).
  - **Combination schemes** (Mihaljevic, Asiacrypt 03).
- *Storage compression techniques*: reducing user storage (Asano 04).
- *Traceable broadcast encryption*: tracing users who forge a decoder (Chor et al, 00).
- *Self-healing broadcast encryption*: allowing recovery of missing group key sent over an unreliable channel (Staddon et al 02).

Key agreement

(the democratic approach...)

Key agreement schemes

In **key agreement schemes**, there is no trusted authority available to assist with the key establishment after the initialisation phase.

The majority of group key agreement schemes are particularly suited to environments where the nature of the communication structure is not known in advance.

For this reason (motivated by potential applications to secure teleconferencing) they are sometimes referred to as **conference key** schemes or **interactive key distribution schemes**.

Key agreement from key (pre)distribution

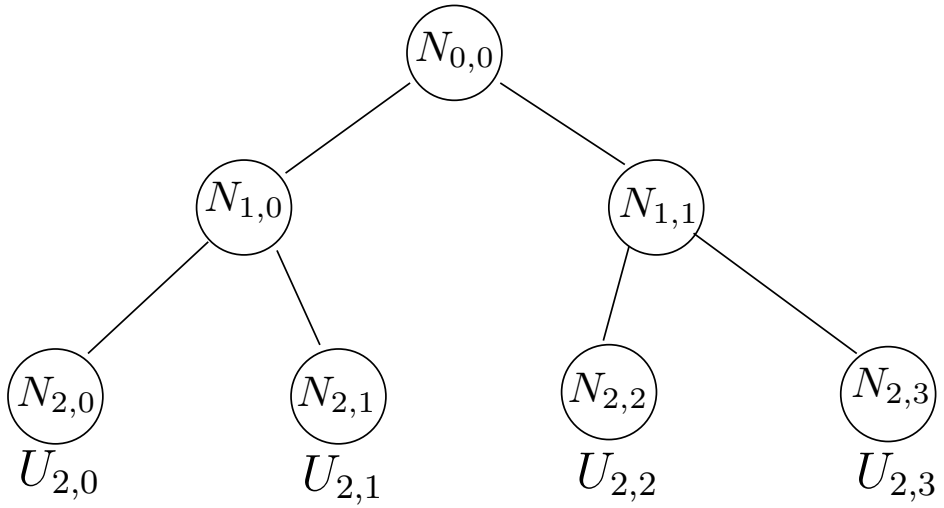
If we have a $(\mathcal{C}, \mathcal{X})$-KPS then we can realise a $(\mathcal{C}, \mathcal{X})$ key agreement scheme as follows:

- Each user stores u_i, as issued in the KPS;
- Users $U_i \in A$ establish group key k_A by utilising secure channels amongst themselves that are protected by the group key k_A^* associated with the KPS.

If we have a $(\mathcal{C}, \mathcal{X})$-KDS then we can realise a $(\mathcal{C}, \mathcal{X})$ key agreement scheme as follows:

- During the initialisation phase, each user is provided with data that allows them to fulfill the role of TA in a $(\mathcal{C}, \mathcal{X})$-KDS;
- Users $U_i \in A$ establish group key k_A by utilising the group keys k_A^{i*} associated with each of the KDSs.

Tree-based Diffie-Hellman (Kim et al 04)



- $U_{2,j}$ generates $k_{2,j}$ and broadcasts $g^{k_{2,j}}$ $(1 \leq j \leq 4)$.
- Users $U_{2,0}$ and $U_{2,1}$ compute $k_{1,0} = g^{k_{2,0}k_{2,1}}$, and $U_{2,2}$ and $U_{2,3}$ compute $k_{1,1} = g^{k_{2,2}k_{2,3}}$. Both $g^{k_{1,0}}$ and $g^{k_{1,1}}$ are then broadcast.
- Group key $k_{0,0} = g^{k_{1,0}k_{1,1}}$ can be computed by all four users.
- Each user stores all keys on the path from their leaf node to the root.

Wireless sensor networks

- Tiny, inexpensive, low-powered **sensors** fitted with wireless transmitters, forming an **ad hoc network**.
- Particularly suited to applications in environments where it is difficult to manually establish a communication network.
- Sensors distributed around the application environment and then attempt to set up a network in order to exchange and return data.
- Actual network topology (defined by a **physical graph**) is not known prior to deployment and is potentially highly dynamic. Thus in many cases we might as well model the physical graph as a random graph.

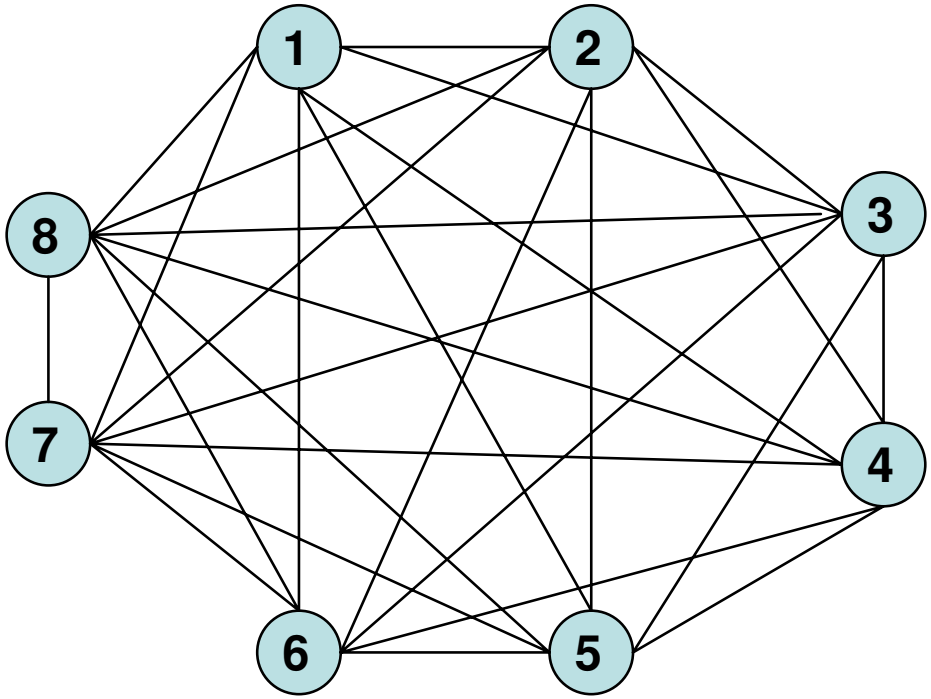
Wireless sensor network scheme

The basic idea behind a **wireless sensor network scheme** for (ideal) communication structure $\mathcal{C}$ is to first establish a $(\mathcal{C}^*, \mathcal{X})$-KPS. We refer to $\mathcal{C}^*$ as the **network communication structure**. When a set $A \in \mathcal{C}$ of sensors requires a group key k_A:

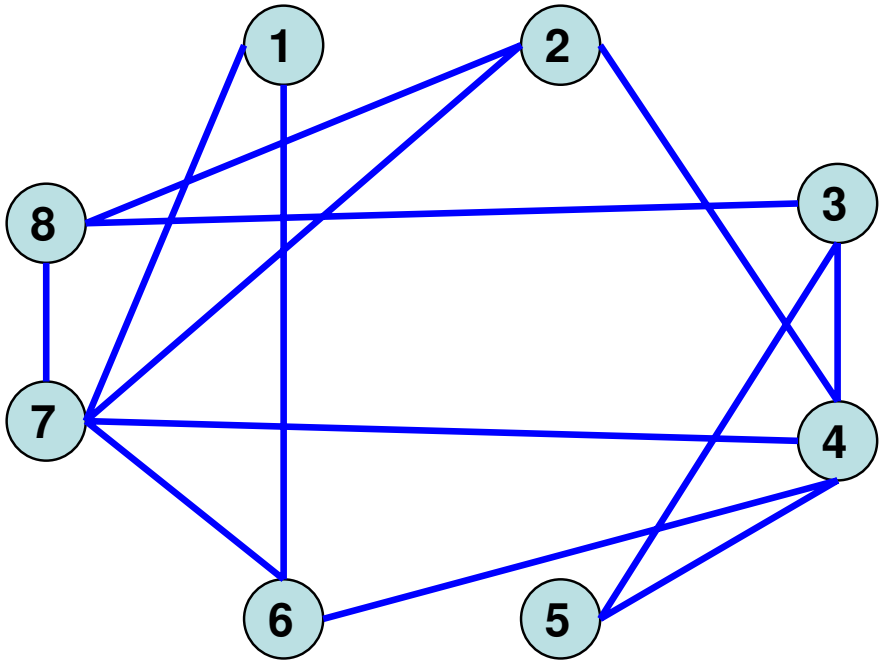
1. if $A \in \mathcal{C}^*$ then they establish k_A using the KPS;
2. if $A \notin \mathcal{C}^*$ then they use some key agreement protocol to establish a key k_A, potentially using other sensors in the network to assist them.

It is reasonable to rely on other sensors to assist in key agreement since the random nature of the physical graph necessitates that nodes typically rely on one another for message transmission services.

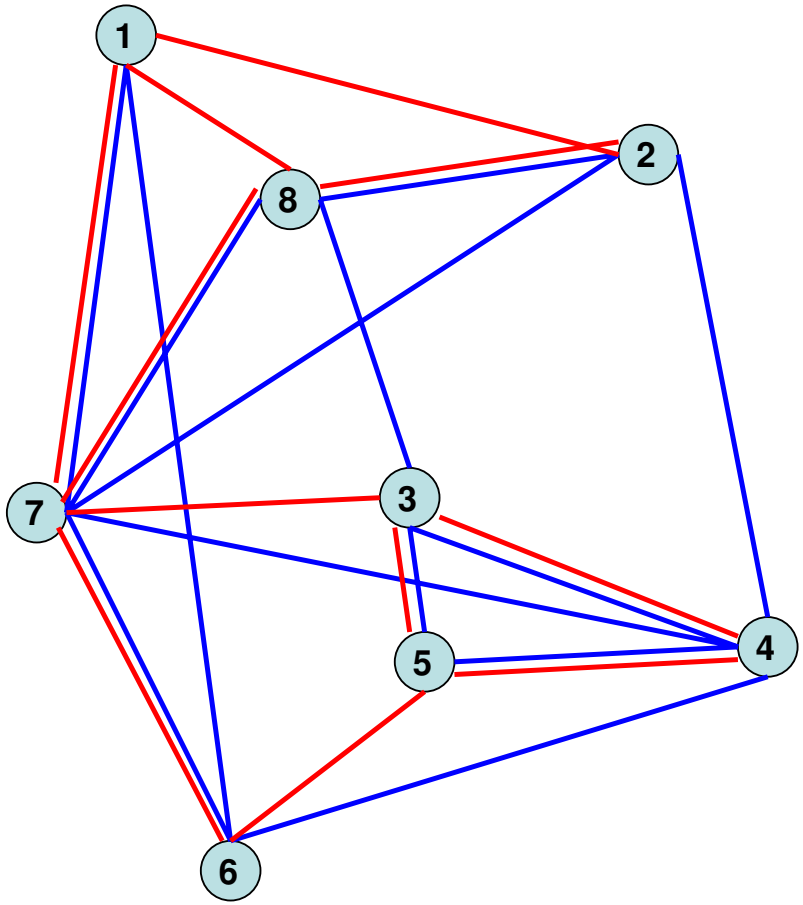
WSN Example: ideal communication structure



WSN Example: network communication structure



WSN Example: physical graph



Key ring WSN schemes

A $(t, n, \mathcal{R})$**-key ring WSN scheme** (KRWSN) is a WSN scheme arising from a key ring predistribution scheme based on $\mathcal{R}$, where the (network) communication structure is

$$\mathcal{C}^* = \{A \subseteq \mathcal{U} \,|\, |A| = t \text{ and } \bigcap_{U_i \in A} u_i \neq \emptyset\}.$$

In other words, a group $U_1, \ldots, U_t$ of t sensors check their public identifier sets $Pub_1, \ldots, Pub_t$ to see if they share any common identifiers.

- If they do then they can establish a group key k_A by applying g to the keys k_i that correspond to the identifiers in $\cap_{j=1}^{t} Pub_j$.
- If not, then they establish the group key by an alternative key agreement mechanism.

Key ring candidates

Suggested candidates for key rings include:

- All subsets of size k (Eschenauer and Gligor, 2002)
- Projective planes (Camtepe and Yener 2004)
- Generalised quadrangles (Camtepe and Yener 2004)
- Transversal designs (Lee and Stinson 2005)
- Common intersection designs (Lee and Stinson 2005)

Comparison of two $(2, n, \mathcal{R})$-KRWSNs

Design	**Projective plane of order q**	**Transversal design TD(k, q)**
Each sensor stores:	$q+1$ keys	k keys
Each pair share:	1 key	0 or 1 keys
Probability of a pair sharing a key	1	$\frac{k}{q+1}$

If we require a WSN with 2400 sensors then:

- using a projective plane of order $q = 49$, each node stores 50 keys
- using a TD$(30, 49)$ each node stores 30 keys and any pair of nodes share a key with probability 0.6

What about nodes that don't share a key?

Set system $(\mathcal{I}, \mathcal{B})$ is a (v, b, r, k, μ)-**common intersection design** if:

1. it has v points, each on r blocks
2. it has b blocks, each with k points
3. any pair of points occur on at most one block
4. for any distinct pair of blocks $B_i, B_j \in \mathcal{B}$ we have:
   $|\{B_k \in \mathcal{B} \,|\, B_i \cap B_k \neq \emptyset \text{ and } B_j \cap B_k \neq \emptyset\}| \geq \mu$.

- Tranversal designs provide one family of examples
- Challenge is to maximise μ while fixing other parameters
- Further investigation of these structures merited

Graph-based WSN schemes

Given that one of our goals in a WSN scheme is to limit the number of hops between sensors who do not share a common key, another sensible design approach is to base the allocation of keys around a virtual **network graph**, whose vertices are sensors and whose edges join sensors who share a common key.

A **graph-based WSN scheme** (GWSN) for a graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ is a pairwise WSN scheme based on an underlying node-based KPS where:

- Each edge $e \in \mathcal{E}$ is associated with a random key k_e;
- $u_i = \{k_e \,|\, U_i \text{ is adjacent to } e\}$;
- $\mathcal{C}^* = \{\{U_i, U_j\} \,|\, U_i \text{ and } U_j \text{ are joined by an edge } e \in \mathcal{E}\}$.

Hybrid WSN schemes

- Extend combinatorial design schemes by adding random blocks. This offers interesting tradeoffs between local connectivity and sensor storage (Camtepe and Yener 2004).
- Randomly merge blocks of combinatorial design schemes, thus creating a key ring with much longer blocks but greater connectivity. This idea thus improves connectivity at the expense of greater sensor storage (Chakrabati et al 2005).

Multisecret sharing schemes

Previous key agreement schemes involved users having to collaborate to construct a group key for practical reasons (such as making key establishment efficient or through restrictions in the connectivity of the network).

In **multisecret sharing schemes** users are forced to collaborate to construct a group key for *security* reasons. This is most likely to happen in applications where the group keys protect sensitive assets that no single user is trusted with the sole authority to access.

A (t, w, λ)-**multisecret threshold scheme** (MTS) is a $(\mathcal{C}, \mathcal{X}, \Gamma)$-multisecret sharing scheme where:

1. $\mathcal{C} = \{A \subseteq \mathcal{U} \,|\, |A| = t\}$;
2. $\mathcal{X} = \{A \subseteq \mathcal{U} \,|\, |A| \leq w\}$;
3. For each $A_i \in \mathcal{C}$, $\Gamma_i = \{X \subseteq A_i \,|\, |X| \geq \lambda\}$.

Closing remarks

(at long last...)

Future directions

- More understanding of the combinatorics of key predistribution (particularly with regard to storage/broadcast tradeoffs).
- More understanding of the extent to which "inefficient" highly structured solutions can be hybridised with "efficient" less structured solutions.
- More understanding of how to obtain efficient solutions in unreliable networks (such as wireless sensor networks).
- More understanding of key establishment schemes for non-threshold communication structures.

Perhaps it's not so surprising that...

- Many key establishment problems have an inherent combinatorial nature
- Combinatorics has played an important role in developing the theory of key establishment
- So many combinatorialists have "sold their souls"...