Encryption

Important Distinction

- Partner share a secret (key)
  $\Rightarrow$ use of fast, symmetric encryption methods
  - classical methods (Caesar, ...) belong to this class
  - One-time pad
  - AES
- Partner have no initial secret
  $\Rightarrow$ use of *public key* cryptography
  (example for 2nd hour falls in this category).

Most applications use a combination of both (e.g. use of public key cryptography to define session key)

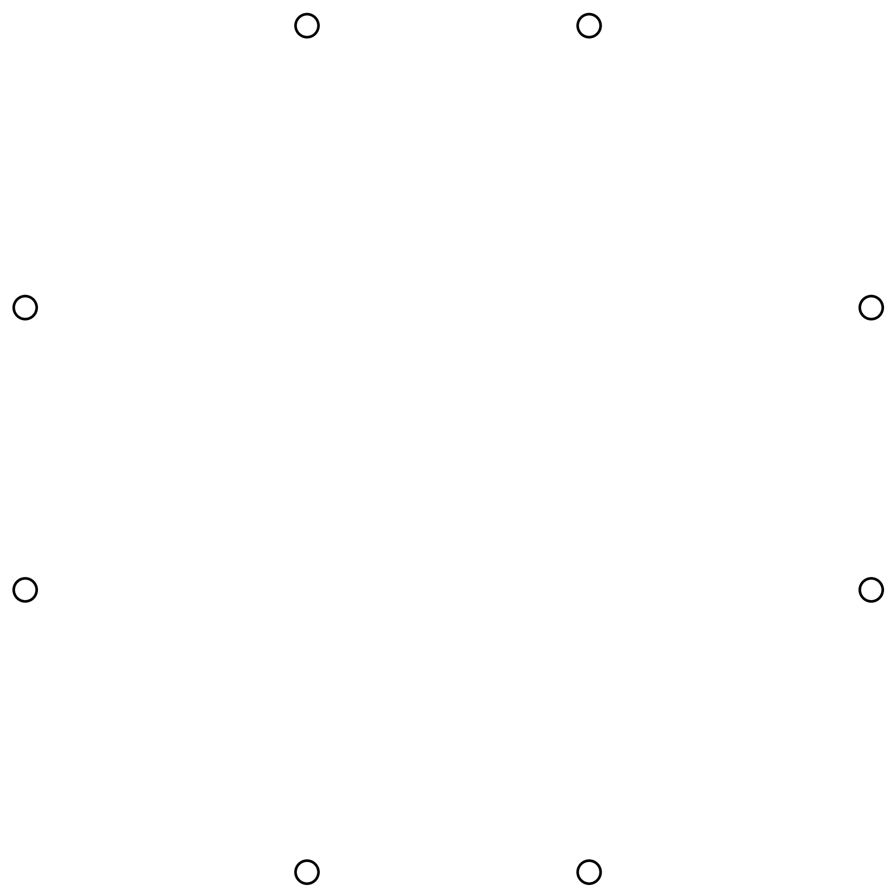
Public Key cryptography

Each user P has two keys

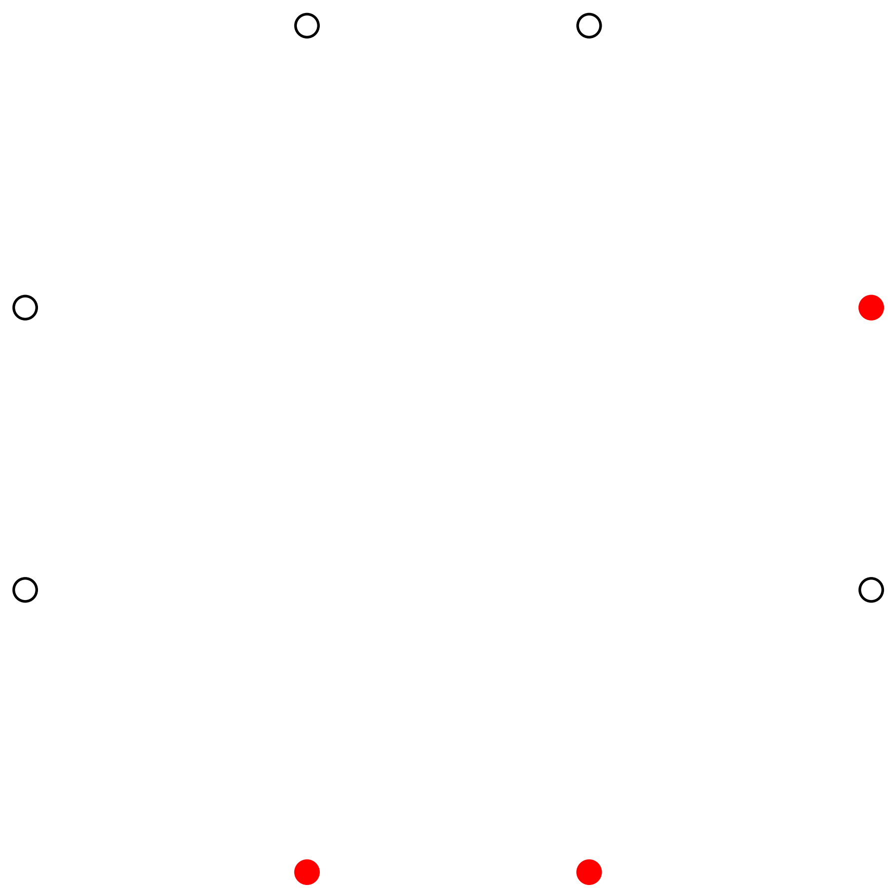
- a private key s_P, chosen by him which never leaves the secure environment,
- a public key p_P, which is made public in a kind of phone book.
- These keys are linked by a (computational) one-way function; the general principles are public.

- Use public key of recipient to encrypt
$\Rightarrow$ only recipient knows secret key and can decrypt.
- Sender uses his private key to sign message
$\Rightarrow$ signature must be due to him as only he knows the secret.

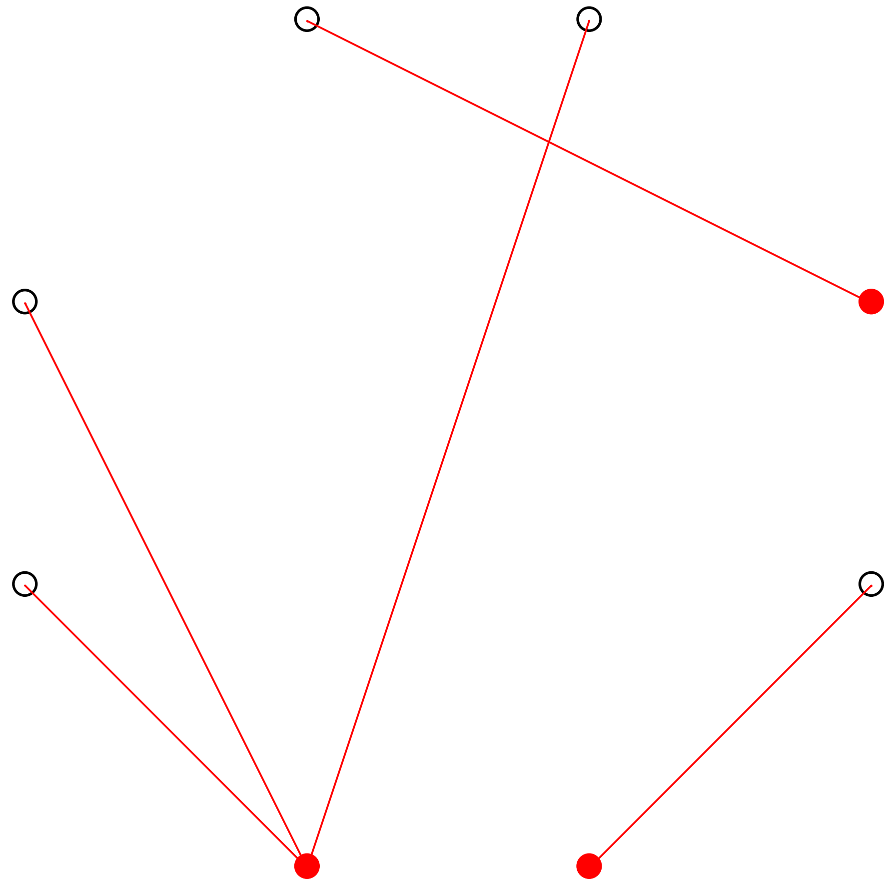
Starting position



Selected nodes = private key

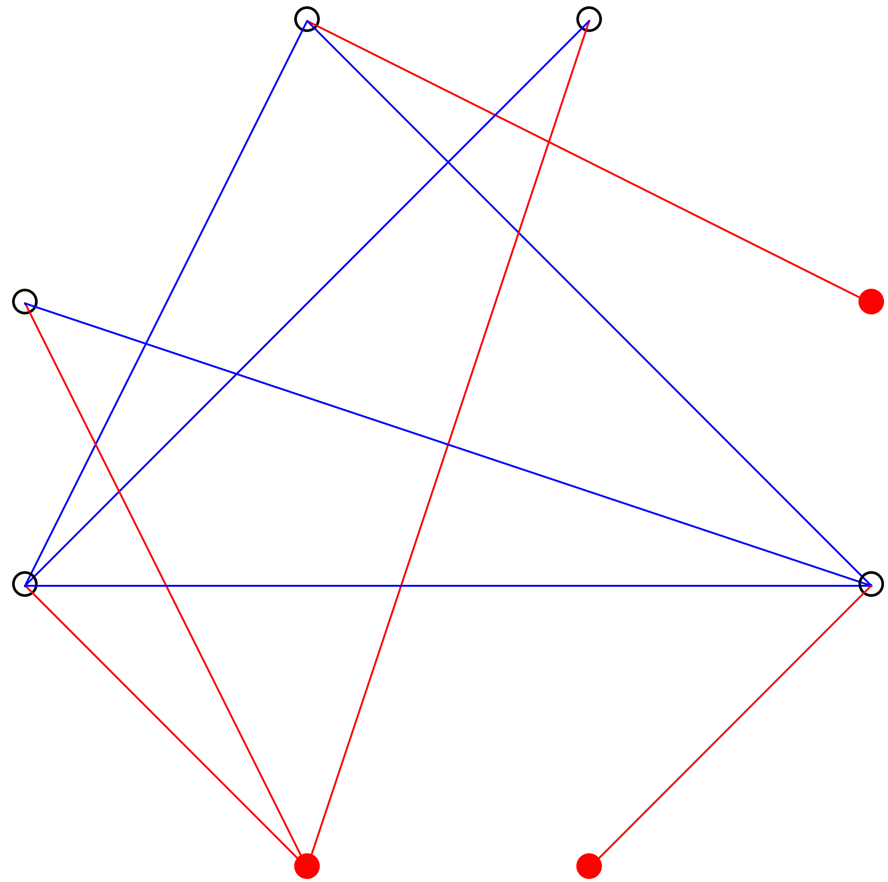


Perfect code – we'll build one



Each node is connected to exactly one selected node.
Perfect code: there exists a selection of nodes so that each node is in the neighborhood of exactly one selected node (a selected node is in its own neighborhood.)

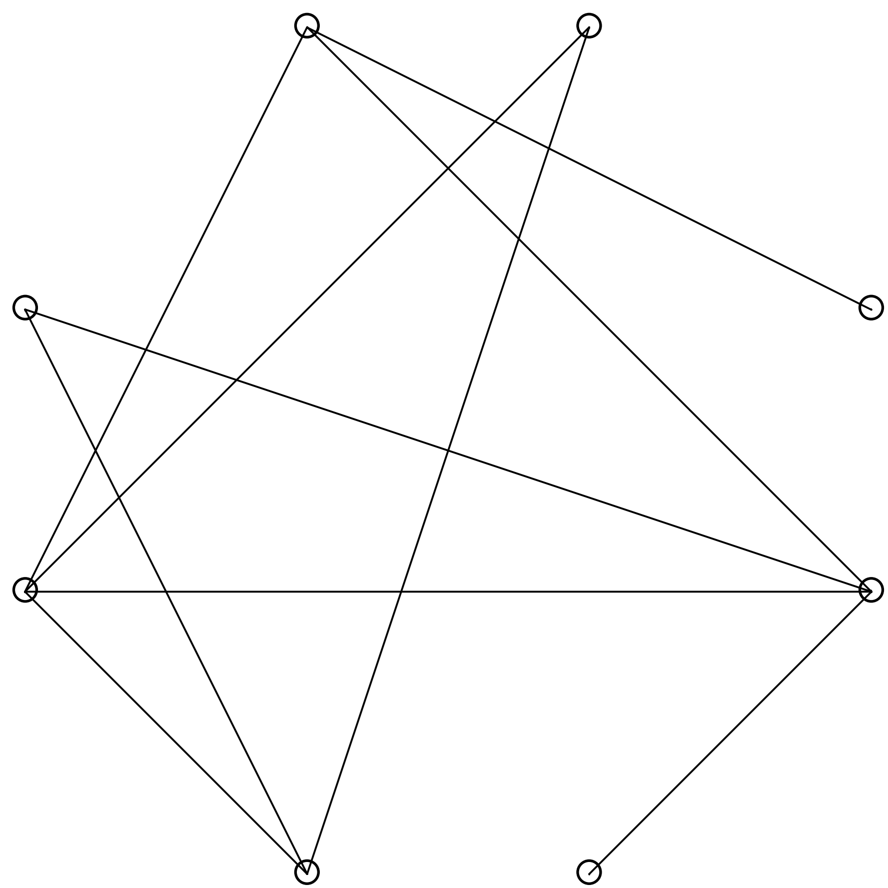
Additional edges



To hide the structure of the selected nodes, further edges are included. These edges must not touch the selected nodes.

This gives a perfect code – proof it!

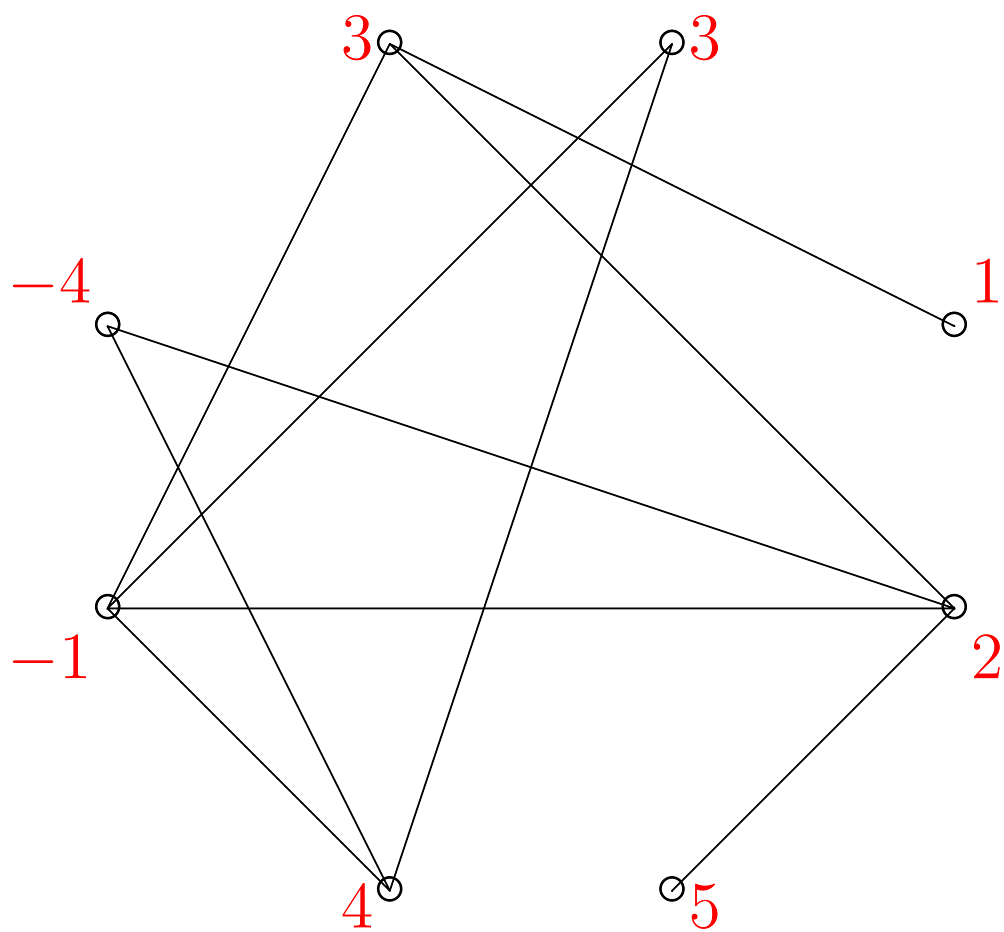
Public key



All edges, no highlighting.

Encryption of $m = 13$

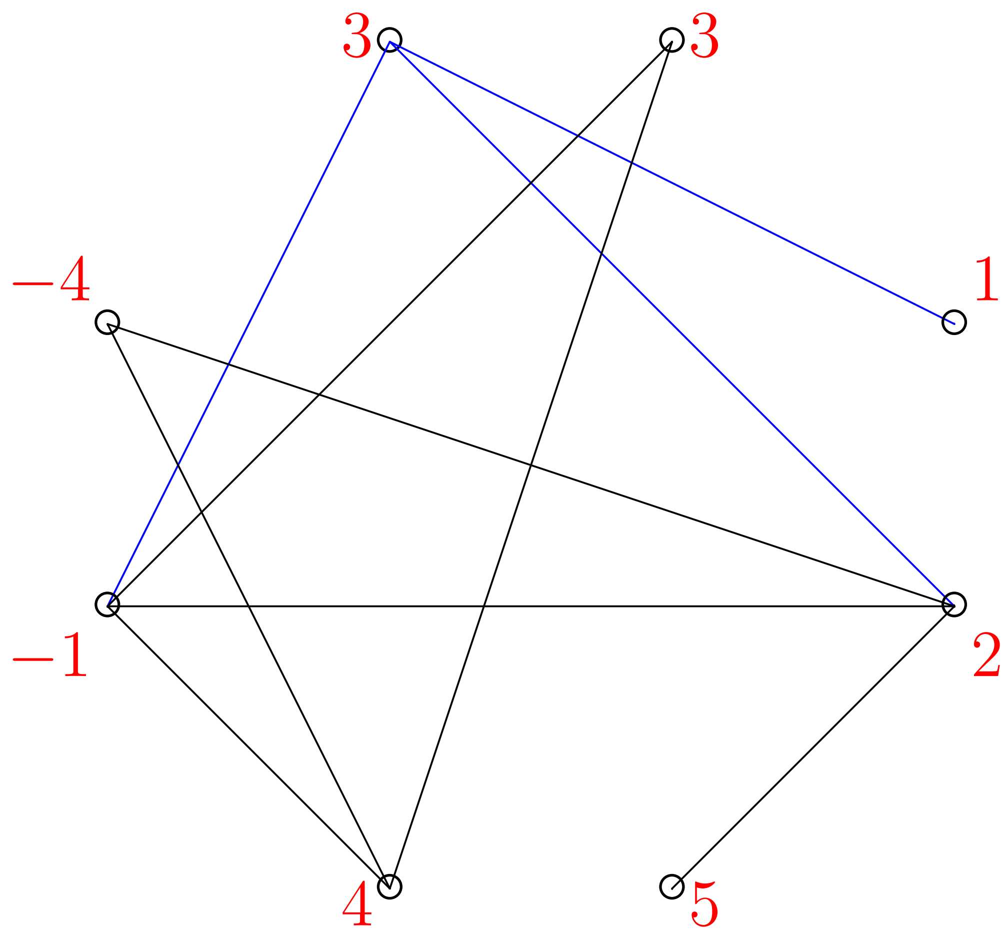
$13 = 1 + 2 + 3 - 4 + 5 + 4 + 3 - 1$. Partition 13, one share per node.



Encryption of $m = 13$

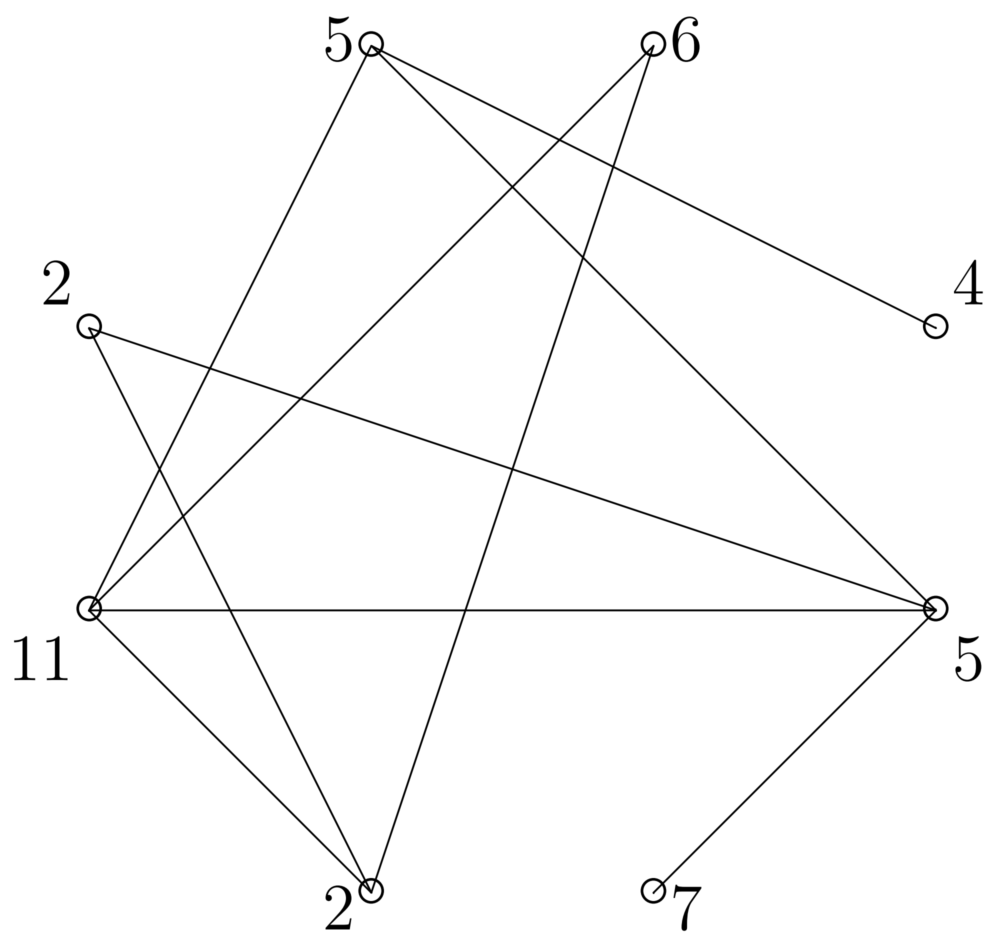
For each node compute the sum of values at all nodes at distance at most 1, i.e. the value at the node itself plus all nodes directly connected to it.

$1 + 2 + 3 - 1 = 5$



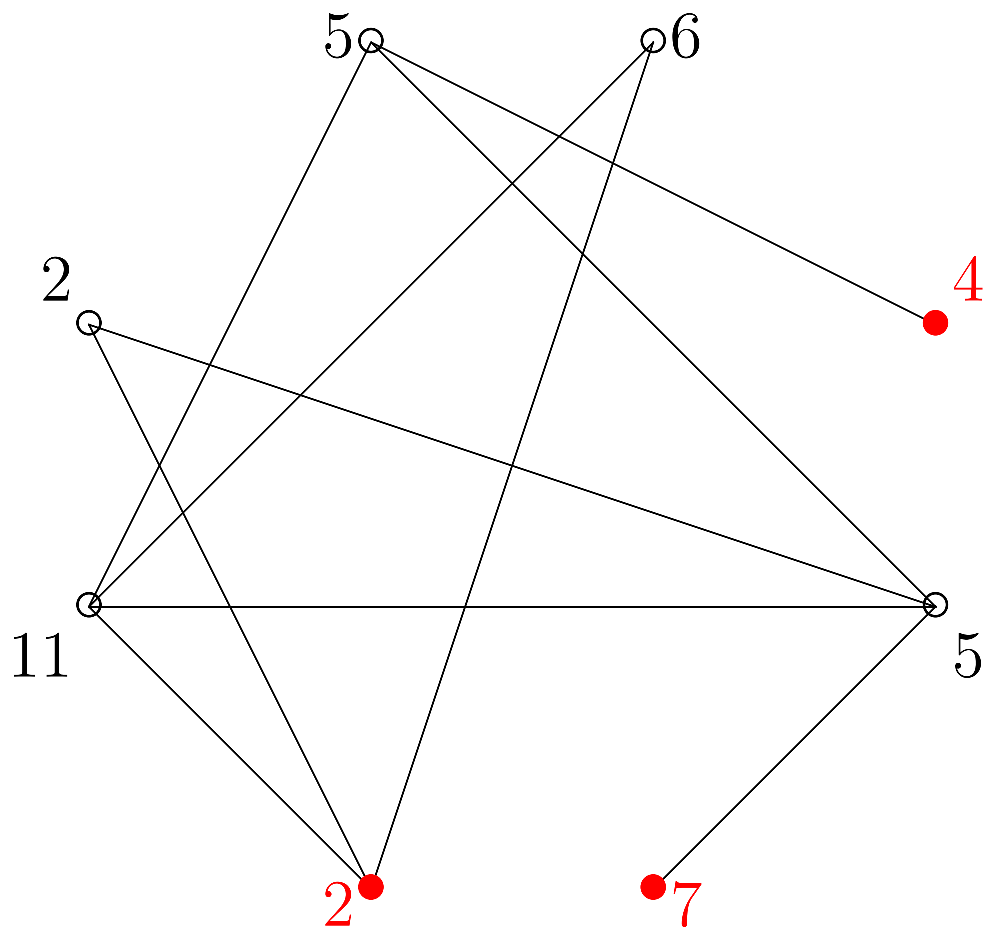
Encrypted message

For each node write the sum computed in the previous step next to it.



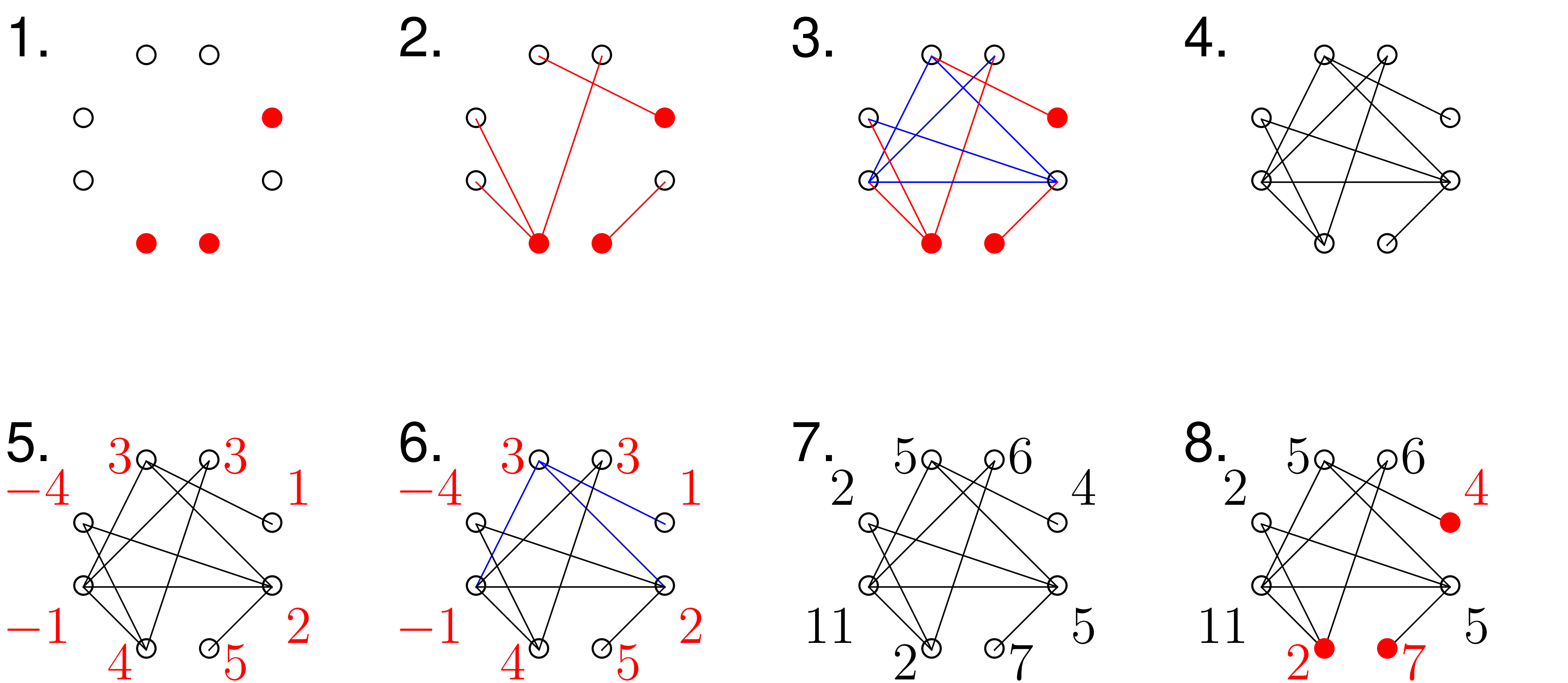
Decryption

Add values at points seleted as secret key.



$4 + 2 + 7 = 13$. Why does this work?

Overview



A: 1. sheet: secret key (1), 2. sheet: public key (4) decryption (8)

intermediate steps (1–3)

B: 1. sheet: computations (5–6) 2.sheet: “black” numbers next to nodes (7)

Why does this system work? Break the examples. Break this for graphs with 1000 nodes.