Cryptography

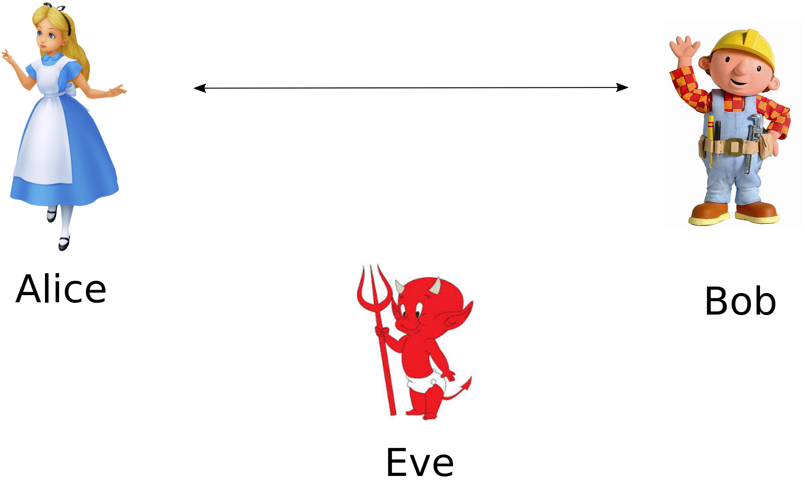
Andreas Hülsing

6 September 2016

Announcements

- Homepage: http://www.hyperelliptic.org/tanja/teaching/crypto16/
- Lecture is recorded ⇒ First row might be on recordings.
- Anything organizational: Ask Tanja on Thursday...

Setting: Alice and Bob want to chat.



Security goals

- **Secrecy,**
- **Integrity,**
- **Authenticity,**
- **Non-repudiation,**
- **(Privacy).**

Security goals

- **Secrecy,** $\Leftarrow$ **We focus on this today**
- **Integrity,**
- **Authenticity,**
- **Non-repudiation,**
- **(Privacy).**

Setting: What about Eve?

Alice

Bob

Eve

Attacker capabilities

- **Passive: Listen.**
- **Active: Intercept & Manipulate. $\Rightarrow$ Change, add, drop content.**

Encryption

Already the Greeks....

Later in Rome

- **Kdoor Fubswr**

Later in Rome

- **Kdoor Fubswr**
- **Hallo Crypto**

Caesar cipher. Also known as ROT3.
"Key table":

a	b	c	d	e	f	g	h	i	j	k	l	m
d	e	f	g	h	i	j	k	l	m	n	o	p
n	o	p	q	r	s	t	u	v	w	x	y	z
q	r	s	t	u	v	w	x	y	z	a	b	c

Symmetric encryption

- Aka. secret key encryption.
- Examples: Caesar, Skytale, . . . , DES, AES.
- **ONE** (secret) key: Stick, rotation, bit string.

Symmetric encryption

- Aka. secret key encryption.
- Examples: Caesar, Skytale, . . . , DES, AES.
- **ONE** (secret) key: Stick, rotation, bit string.

Symmetric encryption

- Aka. secret key encryption.
- Examples: Caesar, Skytale, . . . , DES, AES.
- **ONE** (secret) key: Stick, rotation, bit string.

Semi-formal definition

Symmetric Encryption Scheme

A symmetric encryption scheme $\mathrm{E} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ consists of three PPT algorithms:

- $\mathsf{Kg}(1^n)$: Key generation algorithm. Upon input of security parameter n in unary, outputs a secret key sk.
- $\mathsf{Enc}_{\mathsf{sk}}(m)$: Encryption algorithm. Upon input of a secret key sk and plaintext message m, outputs the encryption / ciphertext c of m under sk.
- $\mathsf{Dec}_{\mathsf{sk}}(c)$: Decryption algorithm. Upon input of a secret key sk and a ciphertext c, outputs the decryption m of c under sk.

Such that:

$$(\forall \mathsf{sk} \leftarrow \mathsf{Kg}(1^n), m) : \mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{sk}}(m)) = m \textbf{ (Completeness)}$$

Remark: Sometimes we use $\mathsf{Enc}(\mathsf{sk}, m) \Leftrightarrow \mathsf{Enc}_{\mathsf{sk}}(m)$.

Semi-formal definition

Symmetric Encryption Scheme

A symmetric encryption scheme $\mathrm{E} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ consists of three PPT algorithms:

$\mathsf{Kg}(1^n)$: Key generation algorithm. Upon input of security parameter n in unary, outputs a secret key sk.

$\mathsf{Enc}_{\mathsf{sk}}(m)$: Encryption algorithm. Upon input of a secret key sk and plaintext message m, outputs the encryption / ciphertext c of m under sk.

$\mathsf{Dec}_{\mathsf{sk}}(c)$: Decryption algorithm. Upon input of a secret key sk and a ciphertext c, outputs the decryption m of c under sk.

Such that:

$$(\forall \mathsf{sk} \leftarrow \mathsf{Kg}(1^n), m) : \mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{sk}}(m)) = m \textbf{ (Completeness)}$$

Remark: Sometimes we use $\mathsf{Enc}(\mathsf{sk}, m) \Leftrightarrow \mathsf{Enc}_{\mathsf{sk}}(m)$.

Security?

- Above definition only functional.
- What does it mean for an encryption scheme to be secure?
- Is Caesar cipher secure? Why not?

Kdoor Fubswr
Hallo Crypto

Semantic security

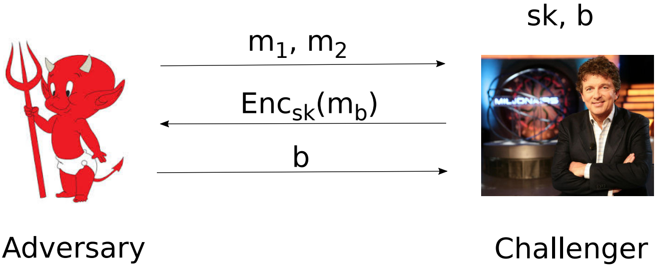
Semantic security: Everything one can learn about a plaintext given its encryption, one can also learn without knowledge of the cipher text.

- Complicated formal definition.
- Hard to work with.
- Technical, equivalent notion: Indistinguishable ciphertexts.

Security Definitions

- Game-based: Adversary vs. Challenger

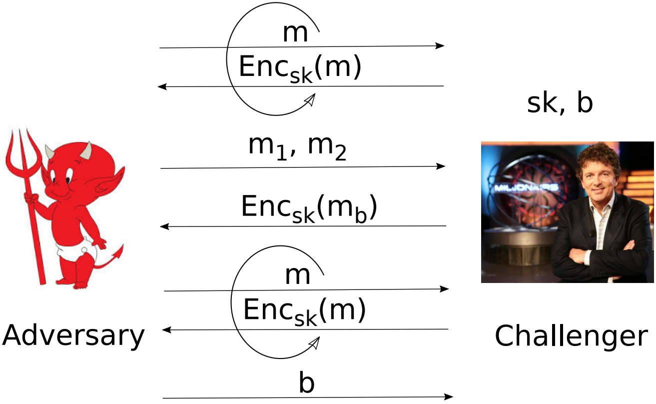
Indistinguishable ciphertexts (IND)



Indistinguishable ciphertexts under choosen plaintext attacks (IND-CPA)

- Adversary might see more ciphertexts than the one she wants to learn more about.
- Attack against Enigma.
- To model worst-case, attacker is allowed to choose plaintexts and learn encryption of those.

Indistinguishable ciphertexts under choosen plaintext attacks (IND-CPA)



Indistinguishable ciphertexts under choosen Ciphertext attacks (IND-CCA)

- Adversary might be able to learn decryptions of ciphertexts other than the target one.
- Users might leak plaintexts corresponding to ciphertexts the adversary saw.
- Practice: Often adversary only learns if a ciphertext is well-formed.
- Model: Additional access to decryption oracle. (Again, worst-case.)
  - Oracle returns either $\mathsf{Dec}_{\mathsf{sk}}(c)$ or $\perp$ if c is no valid ciphertext.

Public-key encryption

- Symmetric encryption is very efficient, but how to share keys?
- Solution: Public-key / Asymmetric encryption.
- Key pair: Public encryption key pk and secret decryption key / private key sk.
- Public key can be published without requiring secrecy.
- Public key can be used to send encryption of (symmetric) secret key.

Semi-formal definition

Asymmetric Encryption Scheme

A symmetric encryption scheme $\mathrm{E} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ consists of three PPT algorithms:

$\mathsf{Kg}(1^n)$: Key generation algorithm. Upon input of security parameter n in unary, outputs a key pair (pk,sk).

$\mathsf{Enc}_{\mathsf{pk}}(m)$: Encryption algorithm. Upon input of a public key pk and plaintext message m, outputs the encryption / ciphertext c of m under pk.

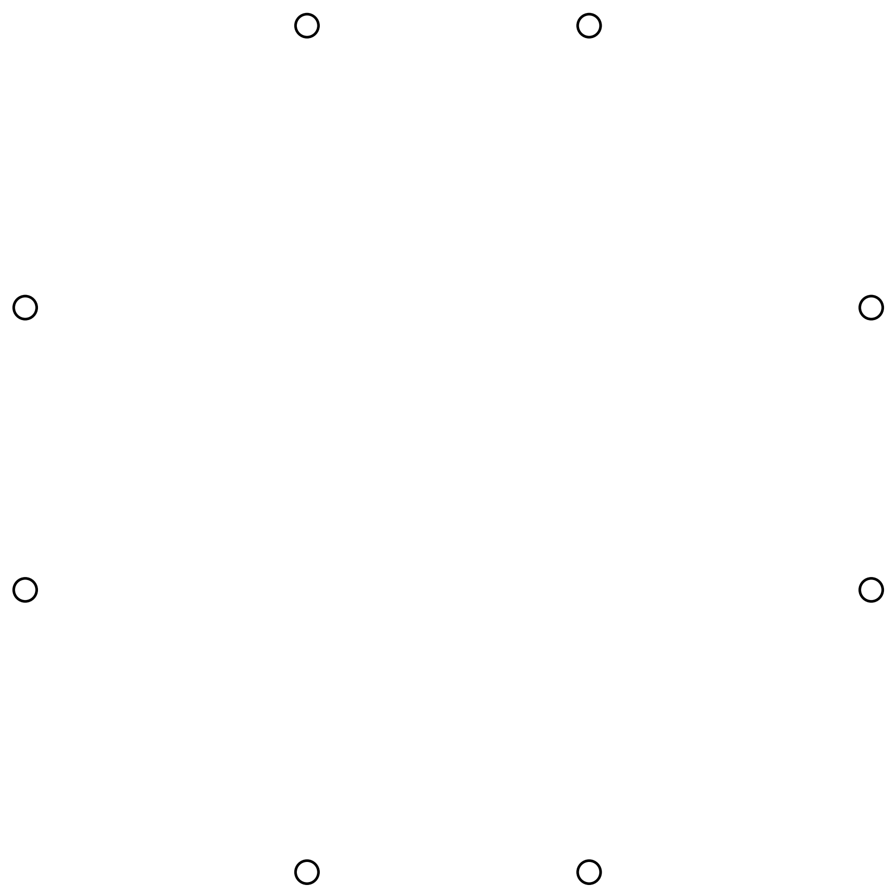
$\mathsf{Dec}_{\mathsf{sk}}(c)$: Decryption algorithm. Upon input of a private key sk and a ciphertext c, outputs the decryption m of c under sk.

Such that:

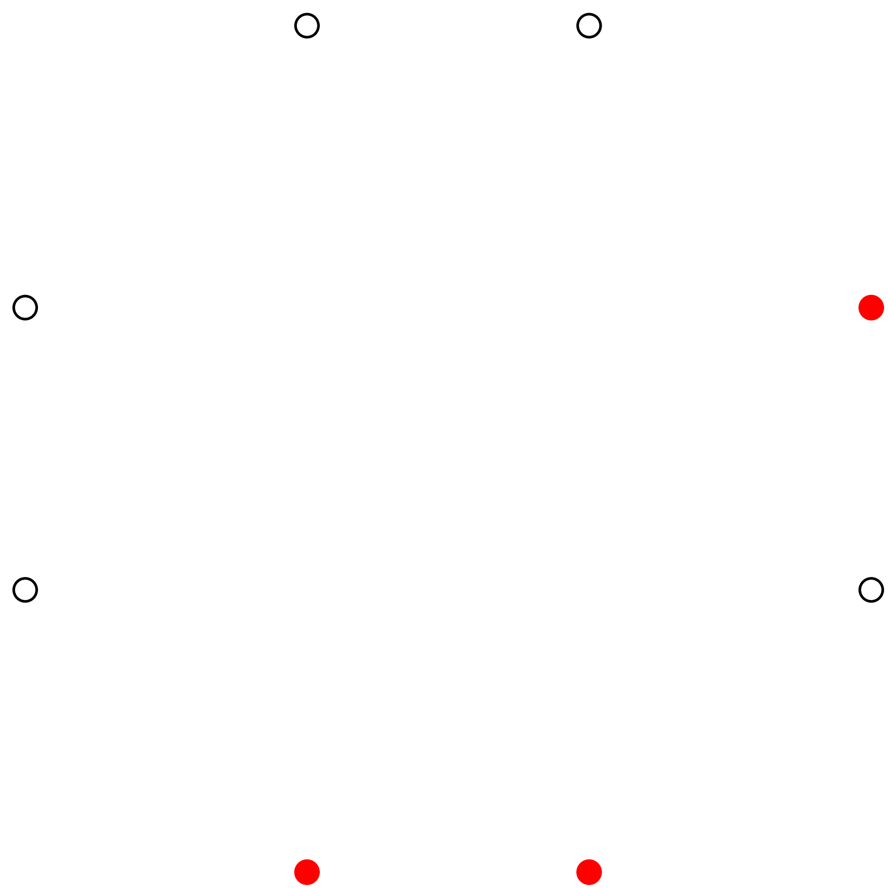
$(\forall(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Kg}(1^n), m) : \mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(m)) = m$ **(Completeness)**

Part II: Break a toy public key encryption scheme.

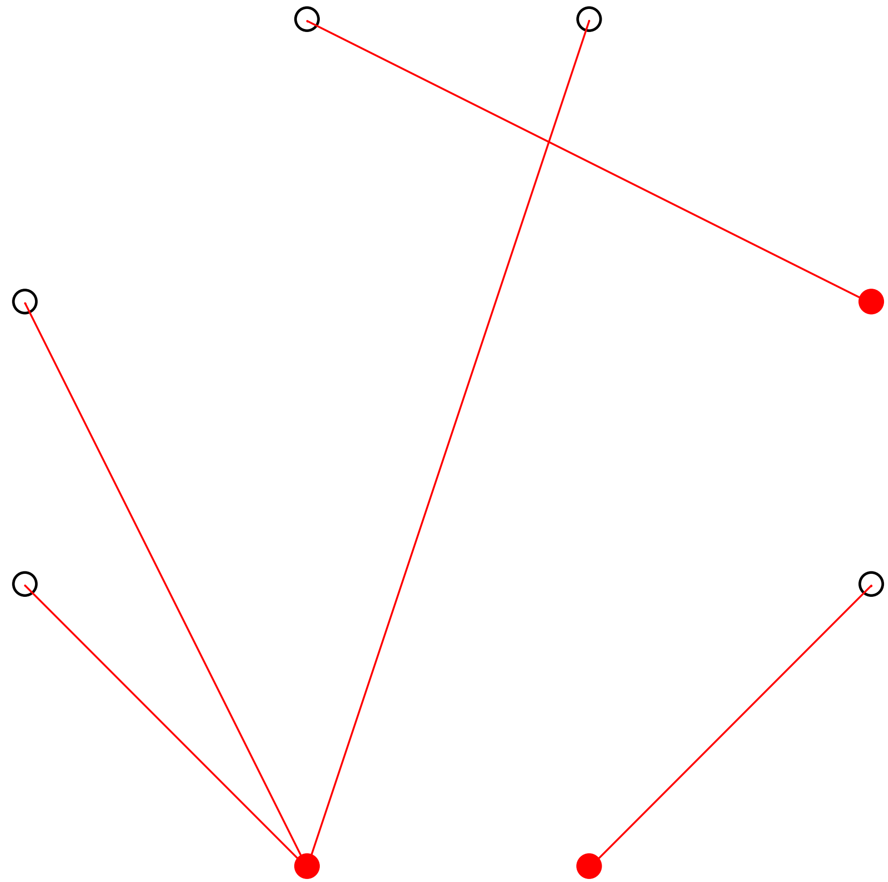
Starting position



Selected nodes = private key

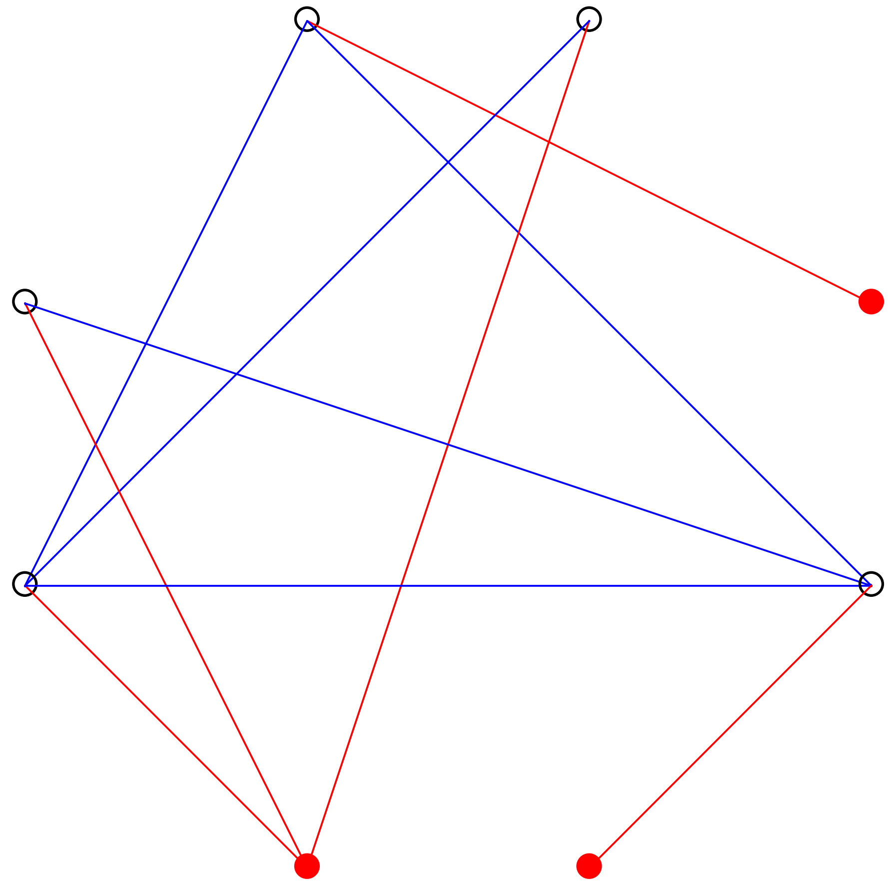


Perfect code – we'll build one



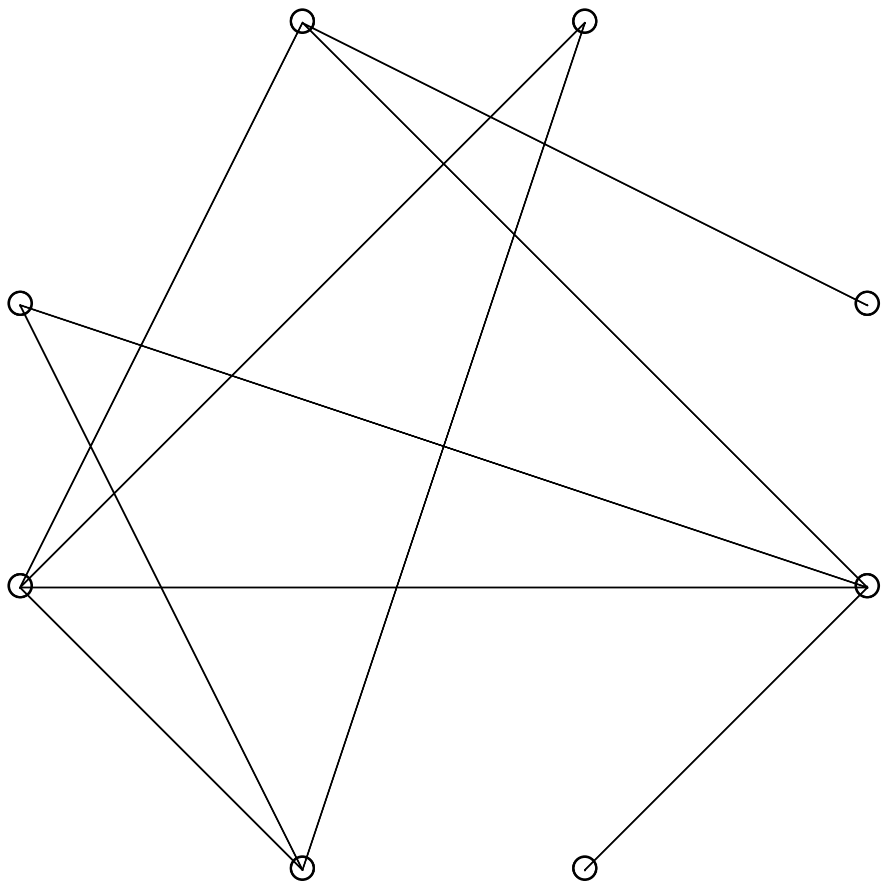
Each node is connected to exactly one selected node.
Perfect code: there exists a selection of nodes so that each node is in the neighborhood of exactly one selected node (a selected node is in its own neighborhood.)

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!

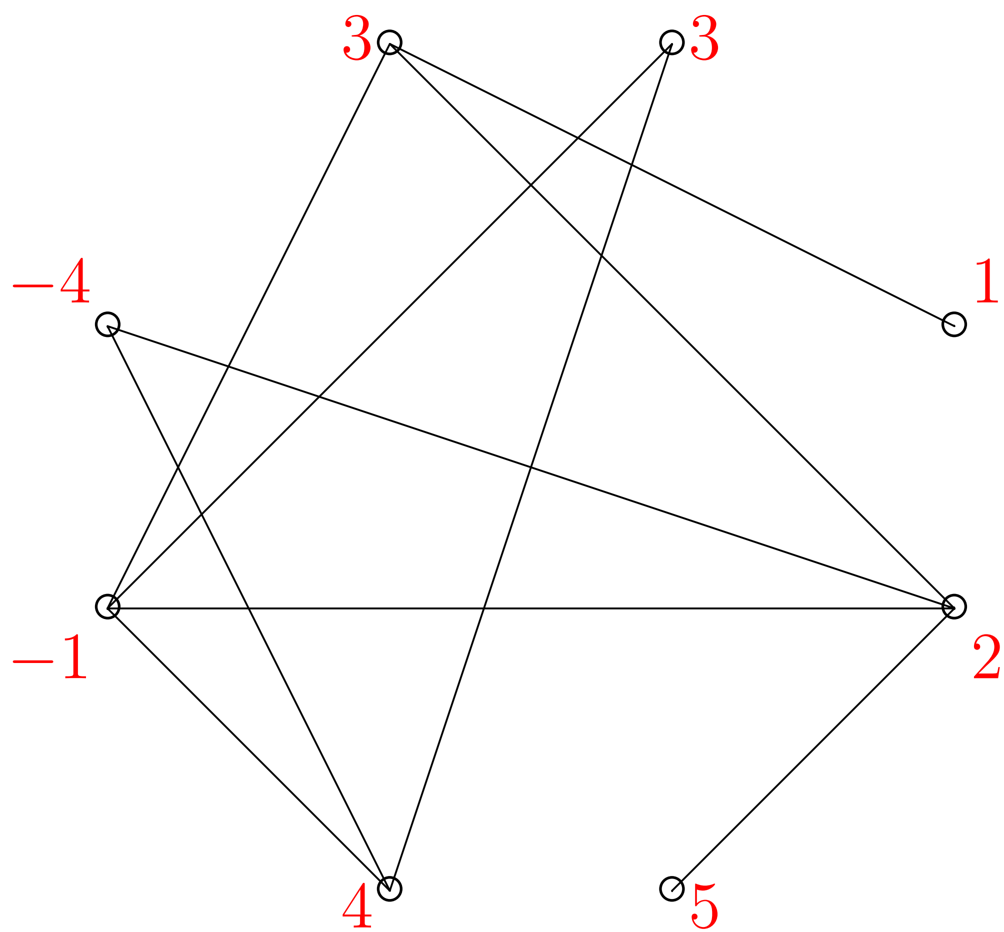
Public key



All edges, no highlighting.

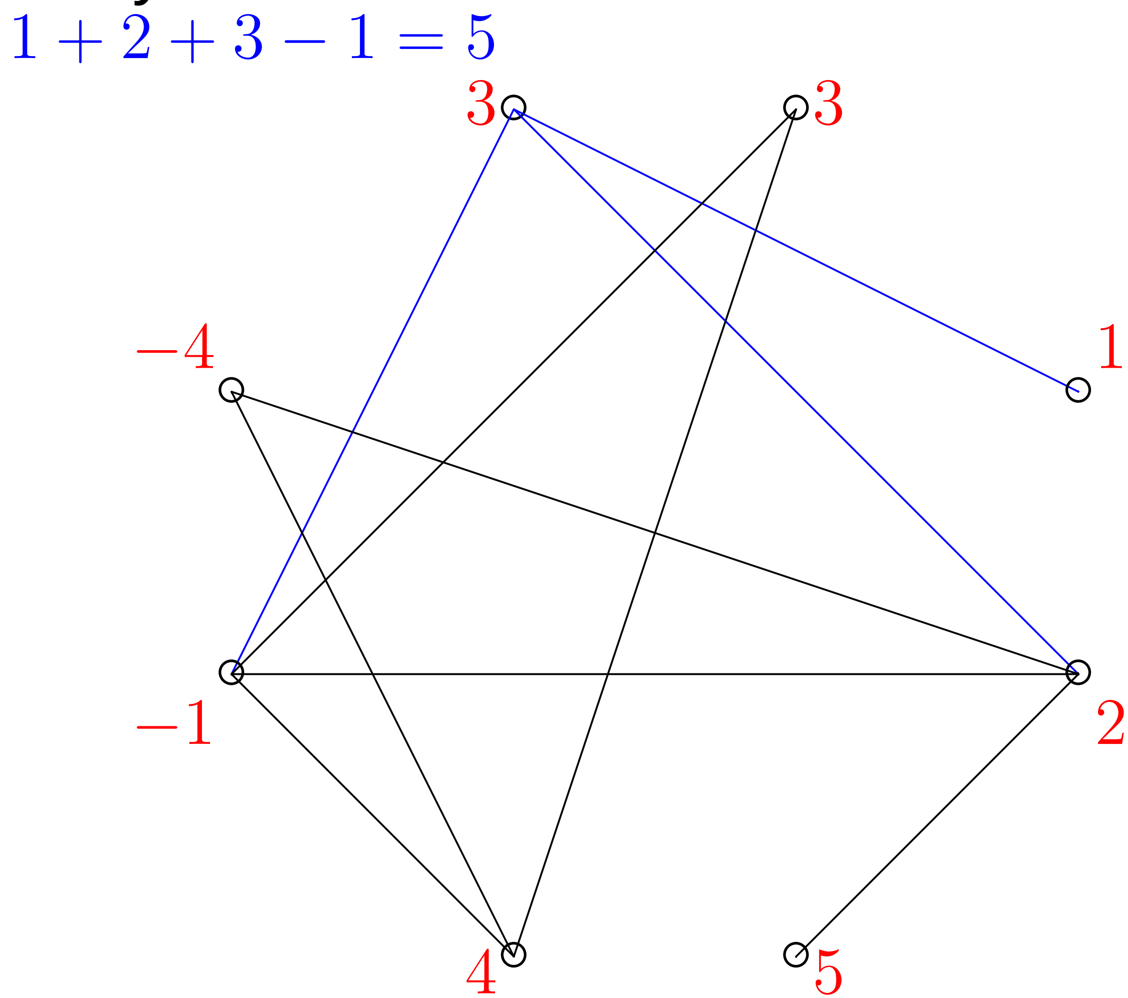
Encryption of $m = 13$

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



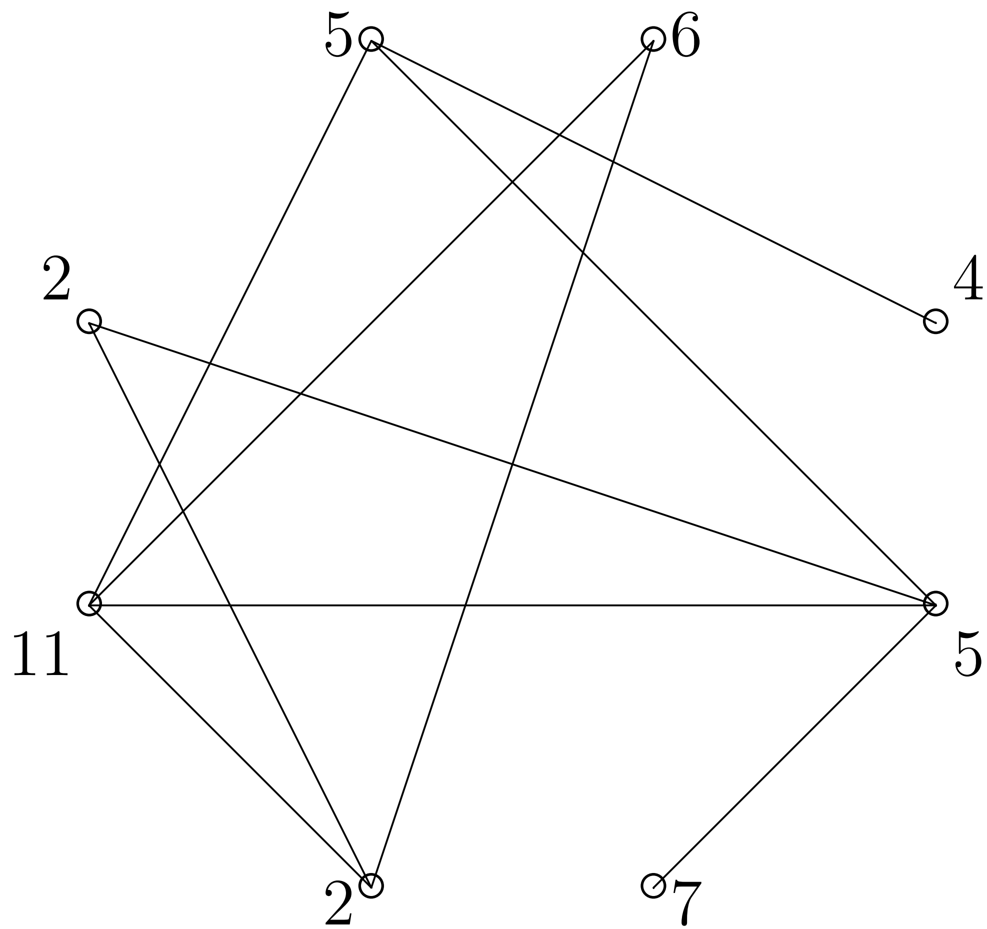
Encryption of $m = 13$

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.



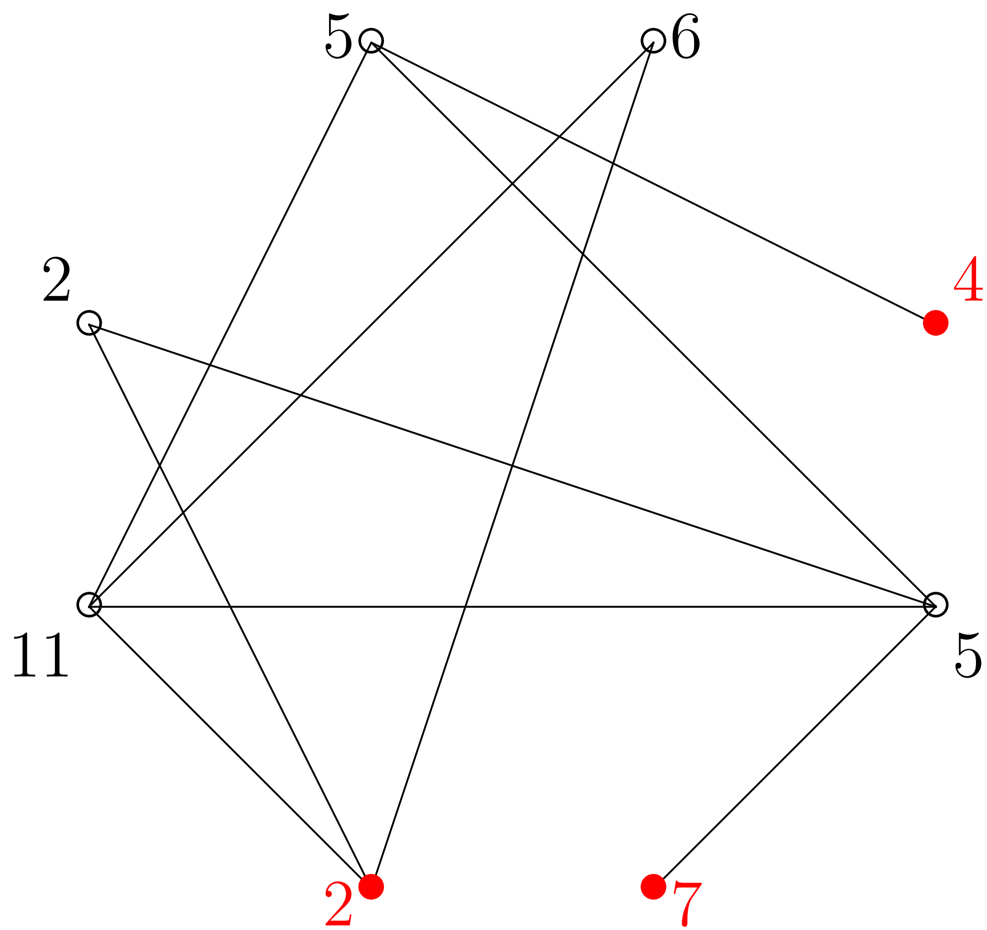
Encrypted message

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



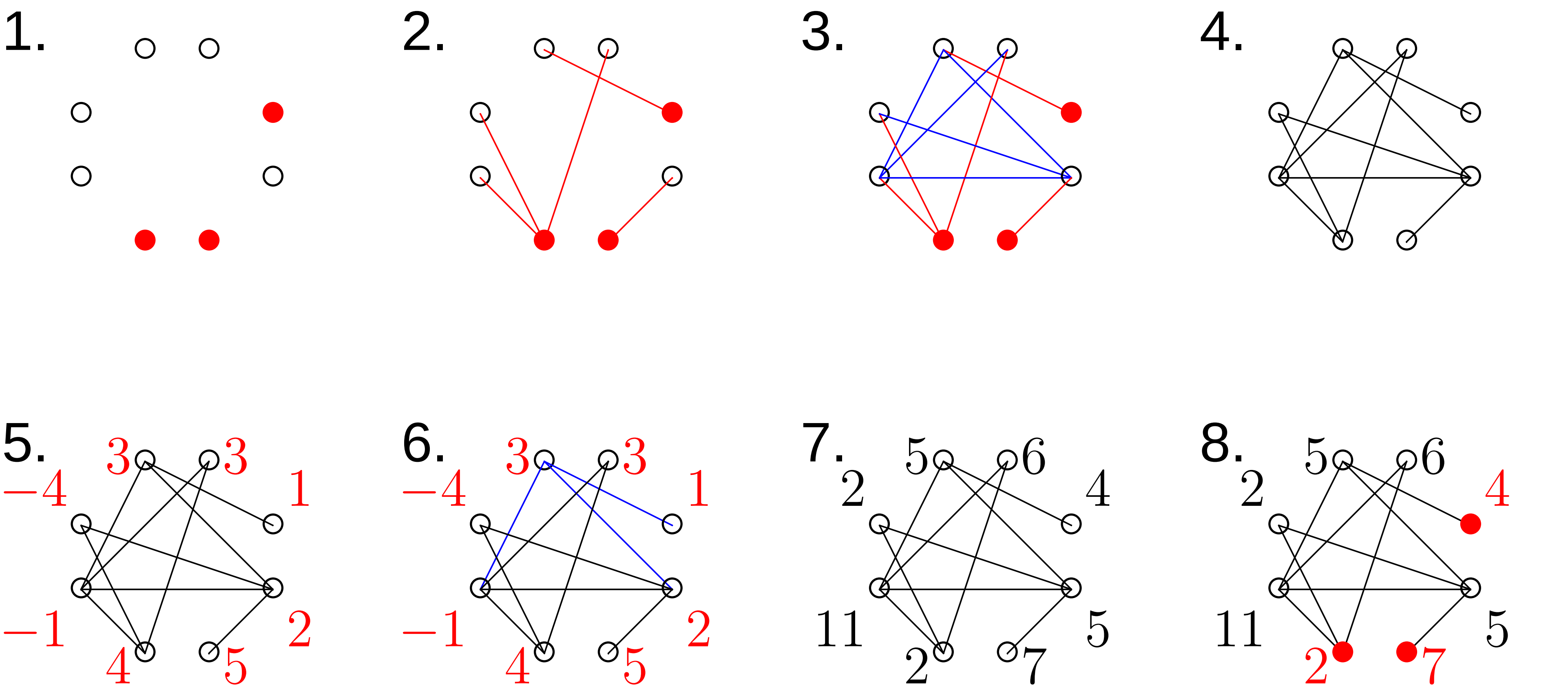
Decryption

Add values at points seleted as secret key.



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

Overview



A: 1. sheet: secret key (1), intermediate steps (1–3) 2. sheet: public key (4) decryption (8)

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.