Peer-to-Peer Networks

14 Security

Christian Schindelhauer
Technical Faculty
Computer-Networks and Telematics
University of Freiburg

Cryptography in a Nutshelf

$P \stackrel{?}{=} NP$

- Symmetric Cryptography
  - AES
  - Affine Cryptosystems
- Public-Key Cryptography
  - RSA
  - ElGamal
- Digital Signatures
- Public-Key-Exchange
  - Diffie-Hellman
- Interactive Proof Systems
  - Zero-Knowledge-Proofs
  - Secret Sharing
  - Secure Multi-Party Computation

Challenge-Response

Blakley 's Secret Sharing

- George Blakley, 1979
- Task
  - n persons have to share a secret

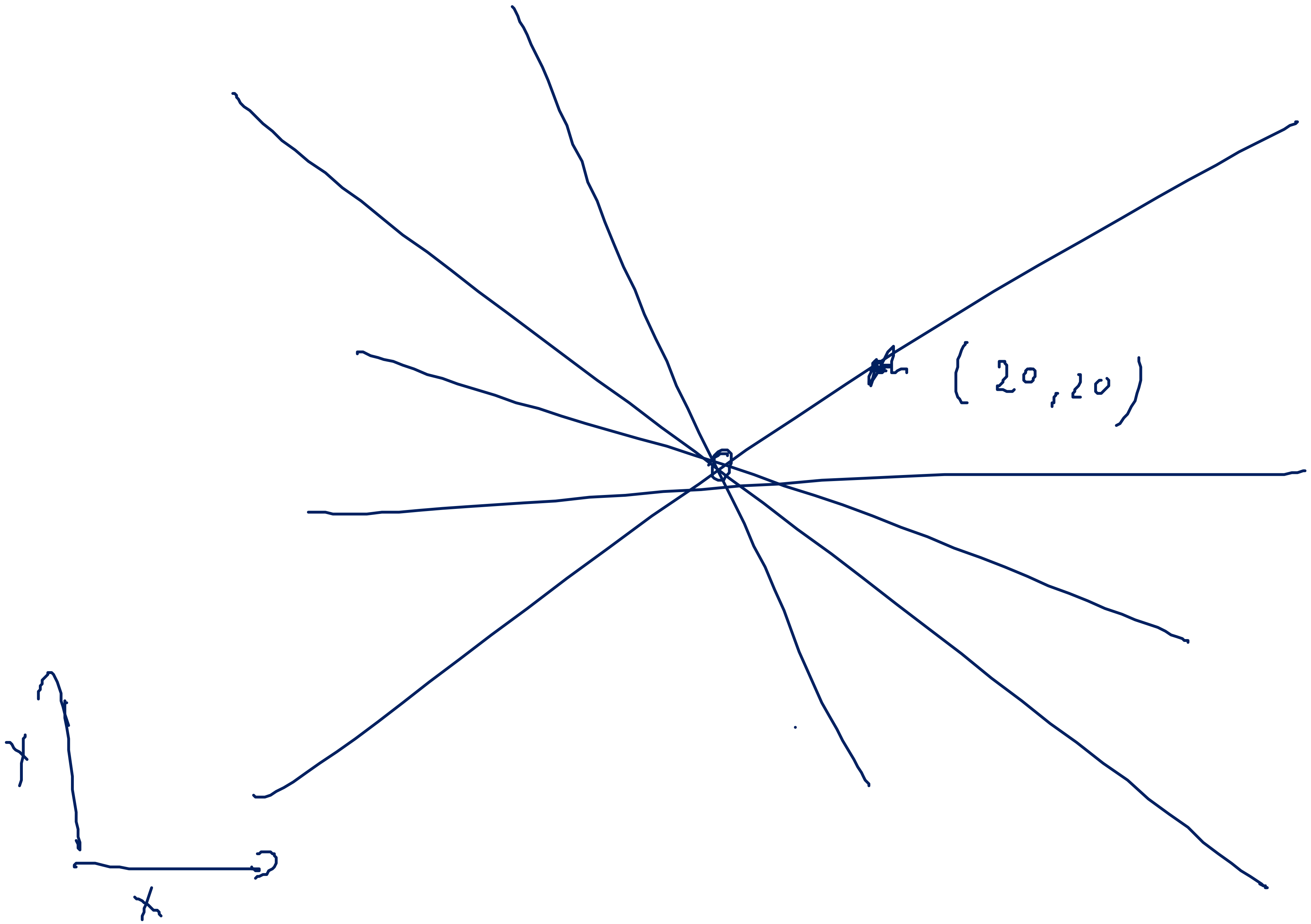
    $n = 5$
    $k = 2$
  - only when k of n persons are present the secret is allowed to be revealed
- Blakley 's scheme
  - in a k-dimensional space the intersection of k non-parallel k-1-dimensional spaces define a point
  - this point is the information
  - with k-1 sub-spaces one gets only a line
- Construction
  - A third (trusted) instance generate for a point n in R^k k non-parallel k-1-dimensional hyper-spaces

Shamir 's Secret Sharing Systems

- Adi Shamir, 1979
- Task
  - n persons have to share a secret s
  - only k out of n persons should be able to reveal this secret
- Construction of a trusted third party
  - chooses random numbers $a_1,...,a_{k-1}$
  - defines
    $$f(x) = \boxed{s} + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{k-1}$$
  - chooses random $x_1, x_2, ..., x_n$
  - sends $(x_i,f(x_i))$ to player i

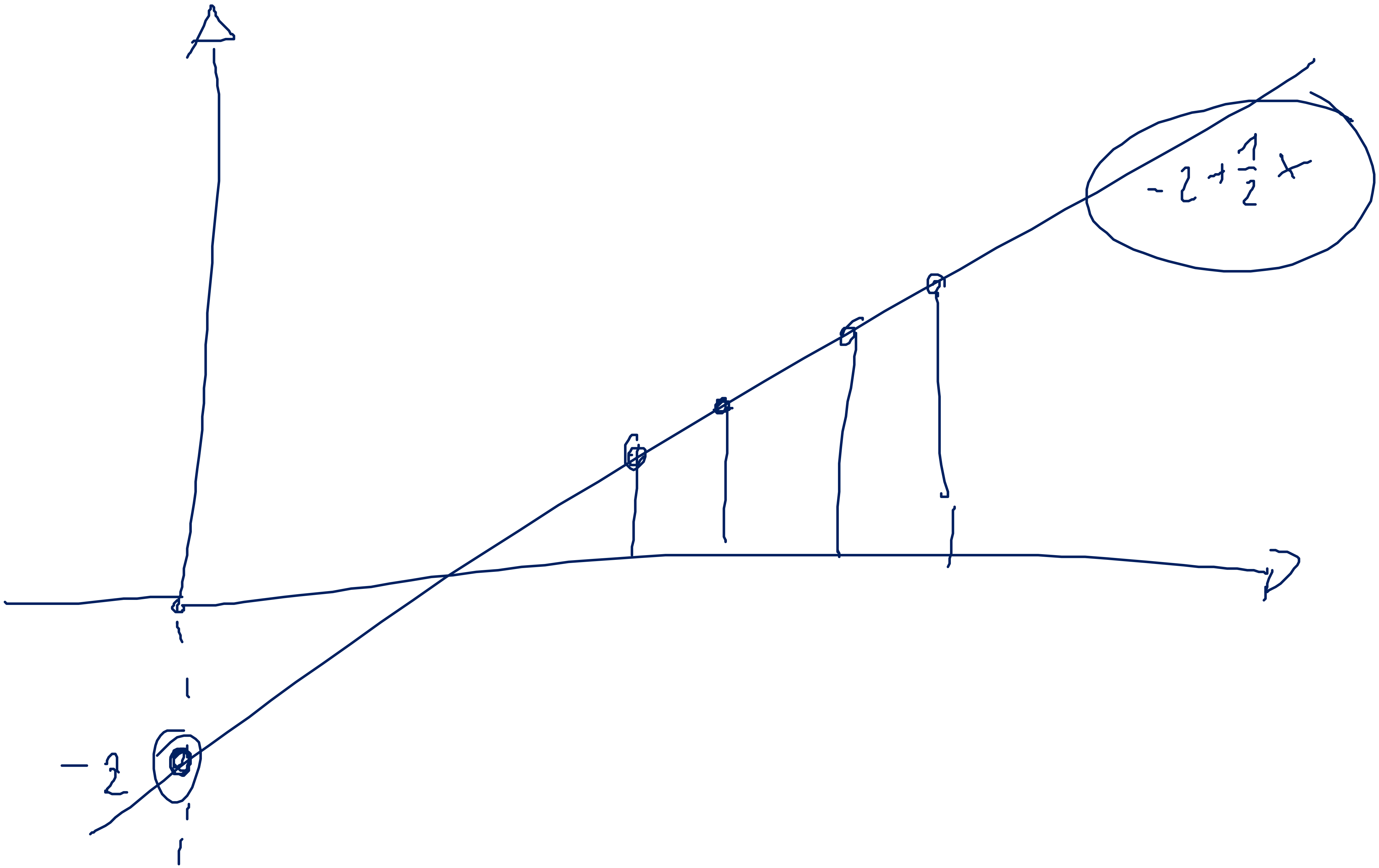
Shamir ‘s Secret Sharing Systems

- If k persons meet
  - then they can compute the function f by the fundamental theorem of algebra
    - a polynomial of degree d is determined by d+1 values
  - for this they exchange their values and compute by interpolation
    - (e.g. using Lagrange polynoms)
- If k-1 persons meet
  - they cannot compute the secret at all
  - every value of s remains possible
- Usually, Shamir ‘s and Blakley ‘s scheme are used in finite fields
  - i.e. Galois fields (known from CRC)
  - this simplifies the computation and avoids rounding errors in the context of floating numbers

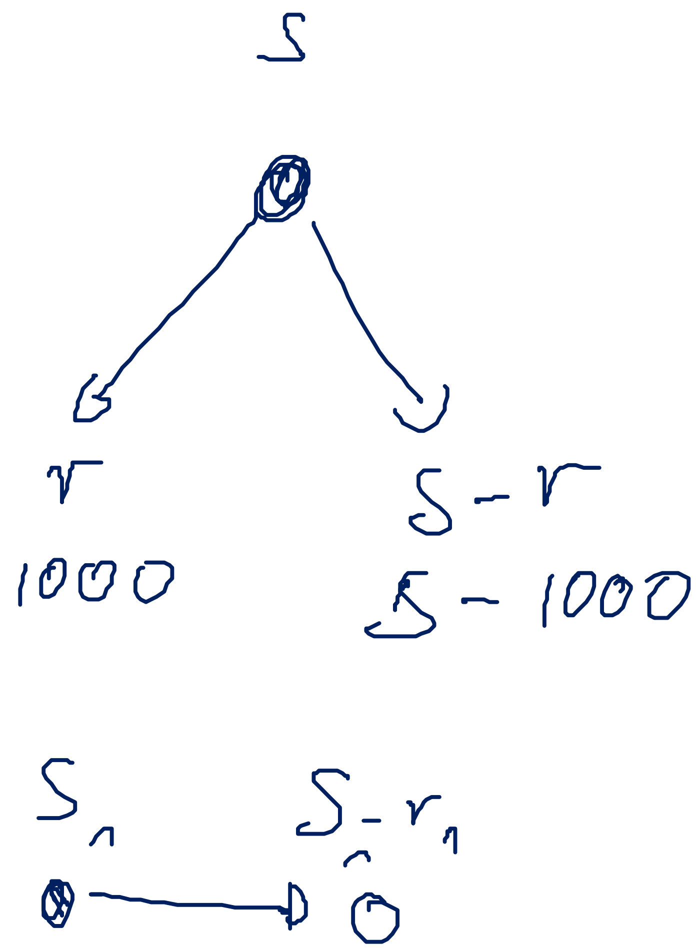

(20,20)
y
x

CoNe Freiburg

$4 = 2$

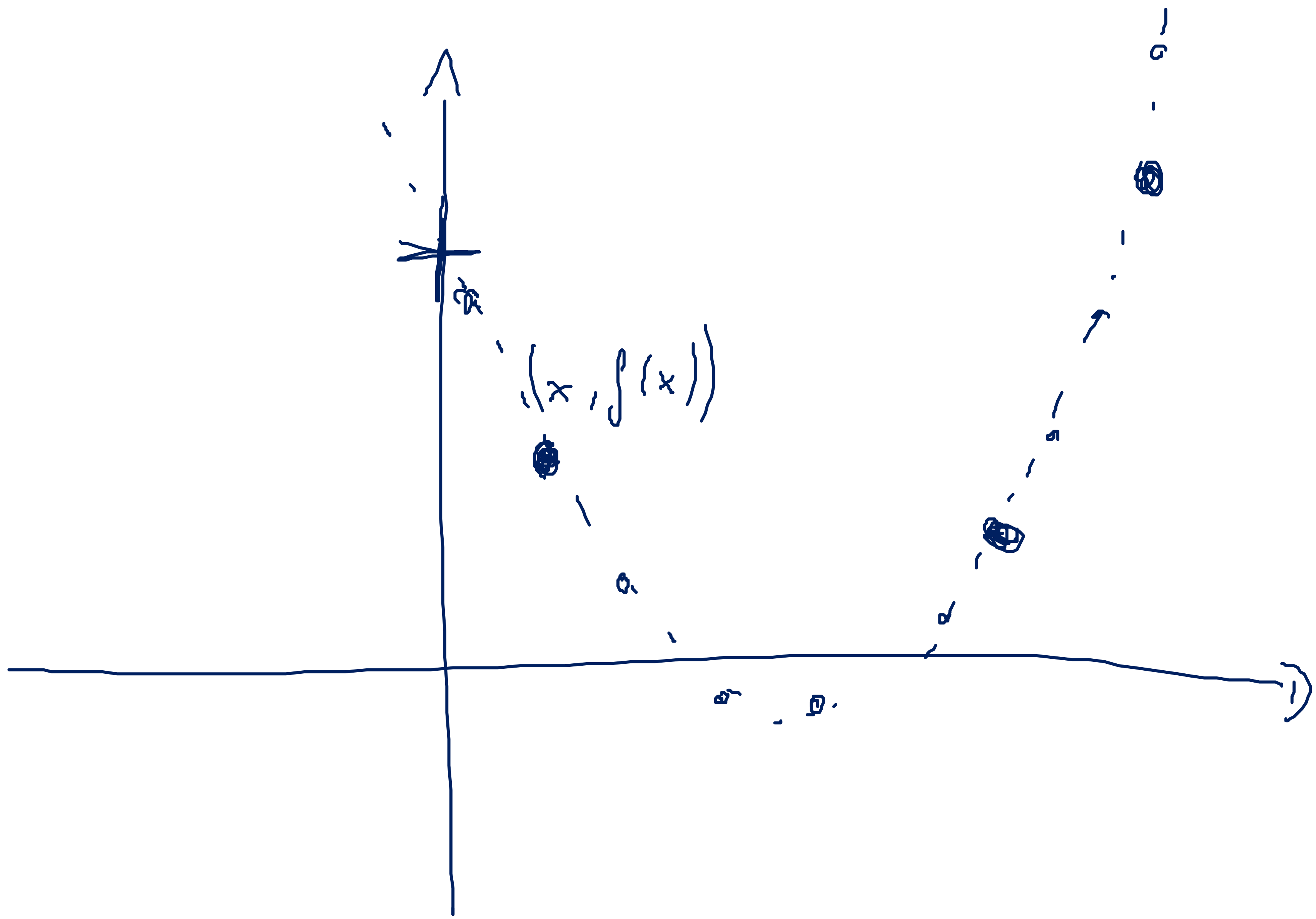


UNI FREIBURG

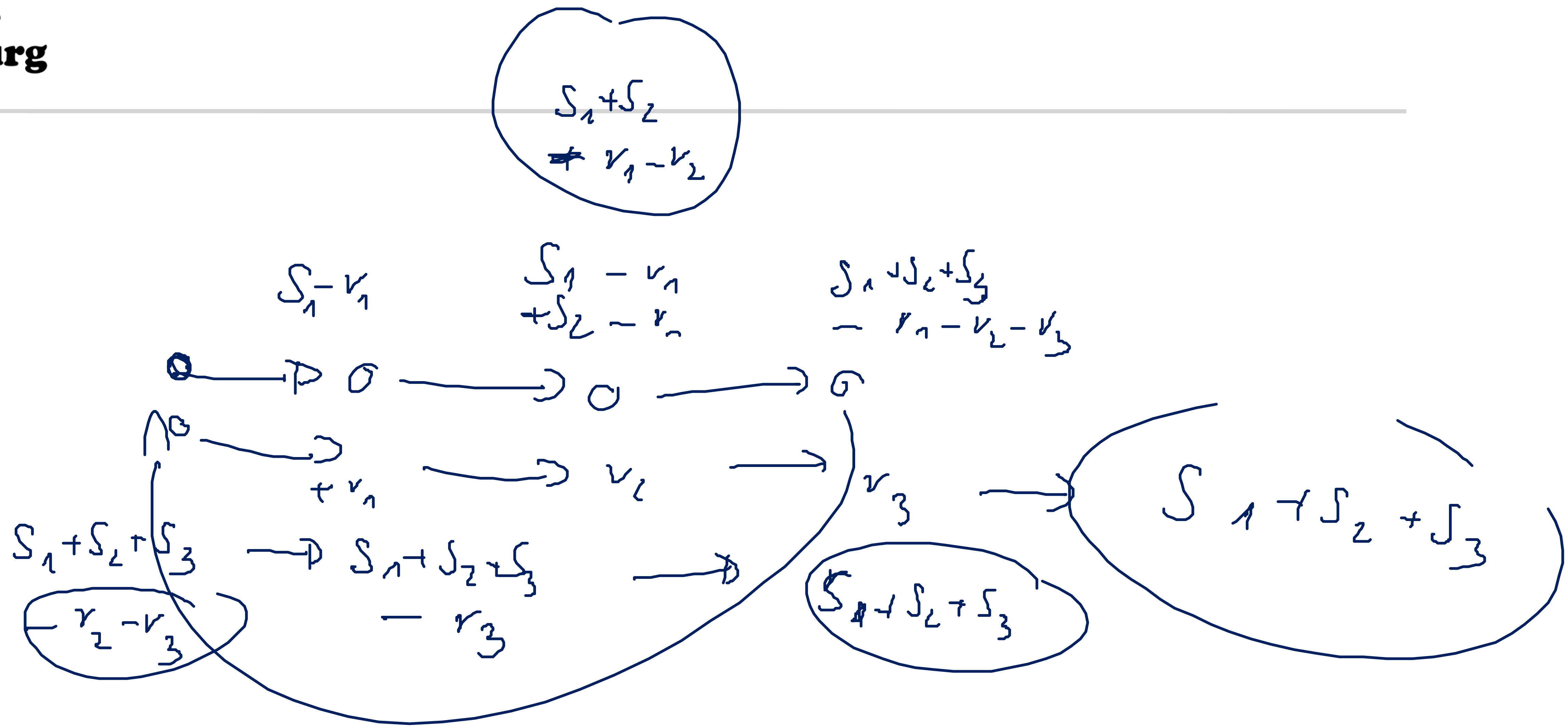

S
r
S - r
1000
S - 1000
S₁
S - r₁

CoNe Freiburg

$$f(x) = 5 + a_1 x + a_2 x^2$$

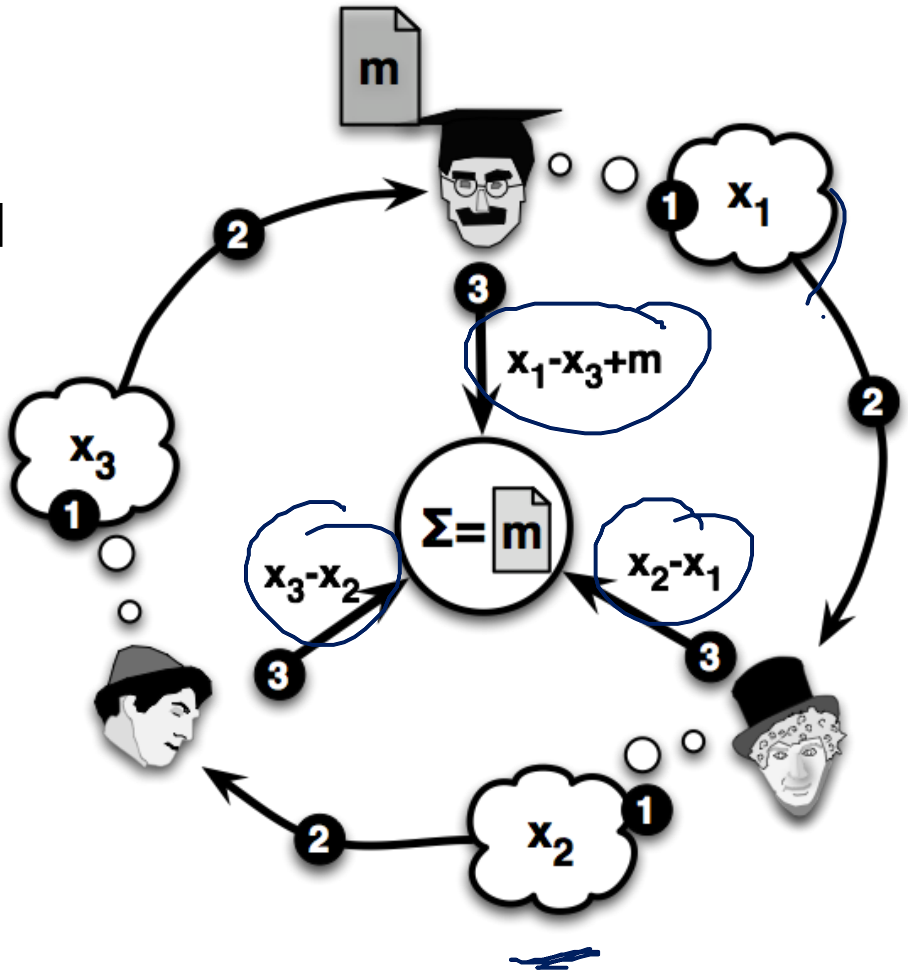


UNI FREIBURG


S1+S2 − v1 − v2
S1 − v1
S1 − v1 +S2 − v2
S1+S2+S3 − v1 − v2 − v3
+ v1
v2
v3
S1+S2+S3 − v2 − v3
S1+S2+S3 − v3
S1+S2+S3
S1+S2+S3

Dining Cryptographers

- Anonymous publications without any tracing possibility
- $n \geq 3$ cryptographers sit at a round table
  - neighbored cryptographers can communicate secretly
- Each peer chooses secret number x_i and communicates it to the right neighbor
- If i wants to send a message m
  - he publishes $s_i = x_i - x_{i-1} + m$
- else
  - he publishes $s_i = x_i - x_{i-1}$
- Now they compute the sum $s=s_1+...+s_n$
  - if s=0 then there is no message
  - else the sum of all messages



Encryption Methods

- Symmetric encryption algorithms, e.g.
  - Feistel cipher
  - DES (Digital Encryption Standard)
  - AES (Advanced Encryption Standard)
- Cryptographic hash function
  - SHA-1, SHA-2
  - MD5
- Asymmetric encryption
  - RSA (Rivest, Shamir, Adleman)
  - El-Gamal
- Digital signatures (electronic signatures)
  - PGP (Phil Zimmermann), RSA

Caesar

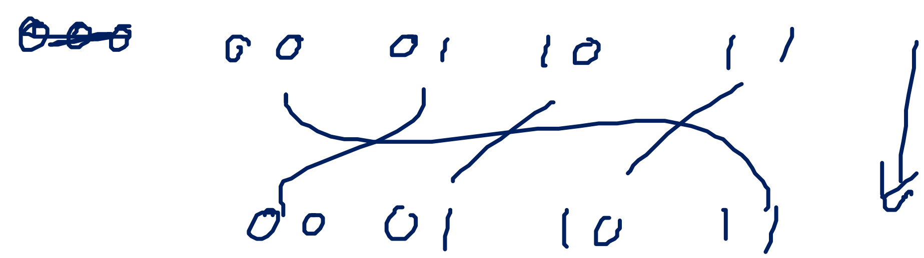
Smart grid

Symmetric Encryption

- E.g. Caesar's code, DES, AES
- Functions f and g, where
  - Encryption f
    - f (key, text) = code
  - Decoding g:
    - g (key, code) = text
- The key
  - must remain secret
  - must be available to the sender and receiver

$f : x \mapsto y$

f bijective

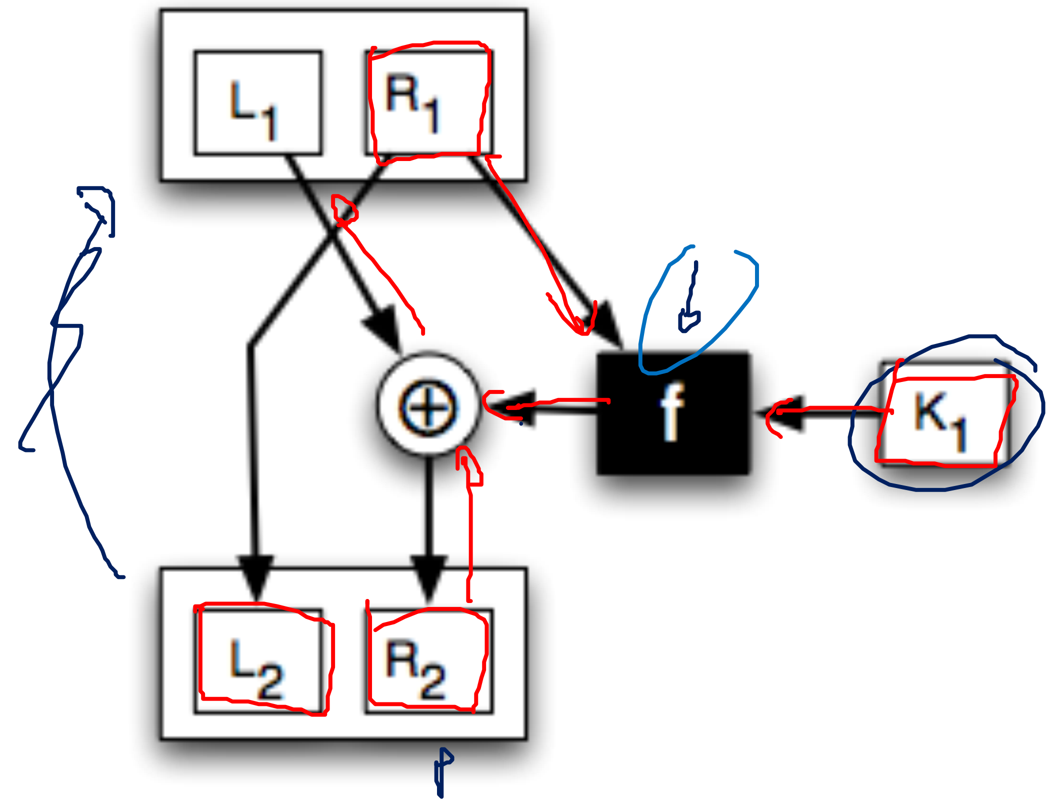


Feistel Chiffre

$x \oplus y \oplus y = x$

$17 \cdot \left\lfloor \frac{R_1 \cdot K_1 + R_1^2 - K_1^3}{107} \right\rfloor$

- Splitting the message into two halves L_1, R_1
  - Keys K_1, K_2, ...
  - Several rounds: Resulting code: L_n, R_n
- encoding
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
- Decryption
  - $R_{i-1} = L_i$
  - $L_{i-1} = R_i \oplus f(L_i, K_i)$
- f may be any complex function



Other Symmetric Codes

- Skipjack
  - 80-bit symmetric code
  - is based on Feistel Cipher
  - low security
- RC5
  - 1-2048 bits key length
  - Rivest code 5 (1994)
  - Several rounds of the Feistel cipher

Digital Encryption Standard

- Carefully selected combination of
  - Xor operations
  - Feistel cipher
  - permutations
  - table lookups
  - used 56-bit key
- 1975 developed at IBM
  - Now no longer secure
  - more powerful computers
  - New knowledge in cryptology
- Succeeded by: AES (2001)

CoNe Freiburg

Advanced Encryption Standard

- Carefully selected combination of
  - Xor operations
  - Feistel cipher
  - permutations
  - table lookups
  - multiplication in GF [2^8]
  - 128, 192 or 256-bit symmetric key

$f(f(x)) = x$

- Joan Daemen and Vincent Rijmen
  - 2001 were selected as AES, among many
  - still considered secure

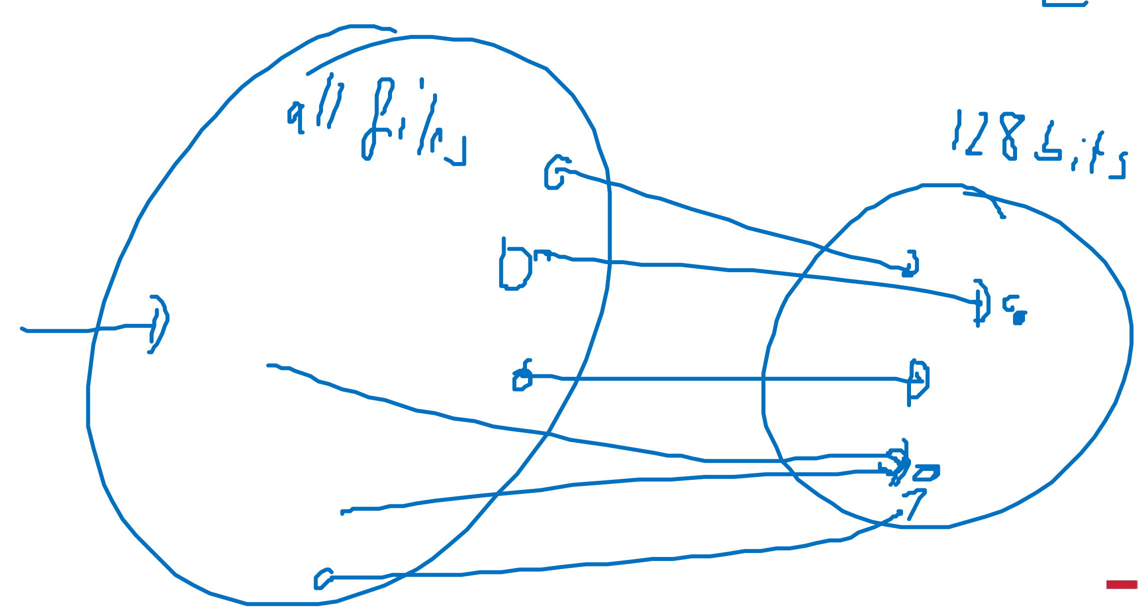
UNI FREIBURG

Cryptographic Hash Function

message digest 5

- E.g. SHA-1, SHA-2, MD5
- A cryptographic hash function h maps a text to a fixed-length code, so that
  - h(text) = code
  - it is impossible to find another text:
    - h(text‘) = h(text) and text ≠ text'
- Possible solution:
  - Using a symmetric cipher

2^{128}



Asymmetric Encryption

3756531072, $\sim \frac{1}{\ln n}$

n

- E.g. RSA, Ronald Rivest, Adi Shamir, Lenard Adleman, 1977
  - Diffie-Hellman, PGP
- Secret key: sk p, q prime
  - Only the receivers of the message know the secret key

$$\boxed{n = p \cdot q}$$

- Public key: pk
  - All participants know this key
- Generated by
  - keygen(sk) = pk

$$(m^s) \bmod n$$

$$s \cdot p \equiv 1 \bmod (p-1)(q-1)$$

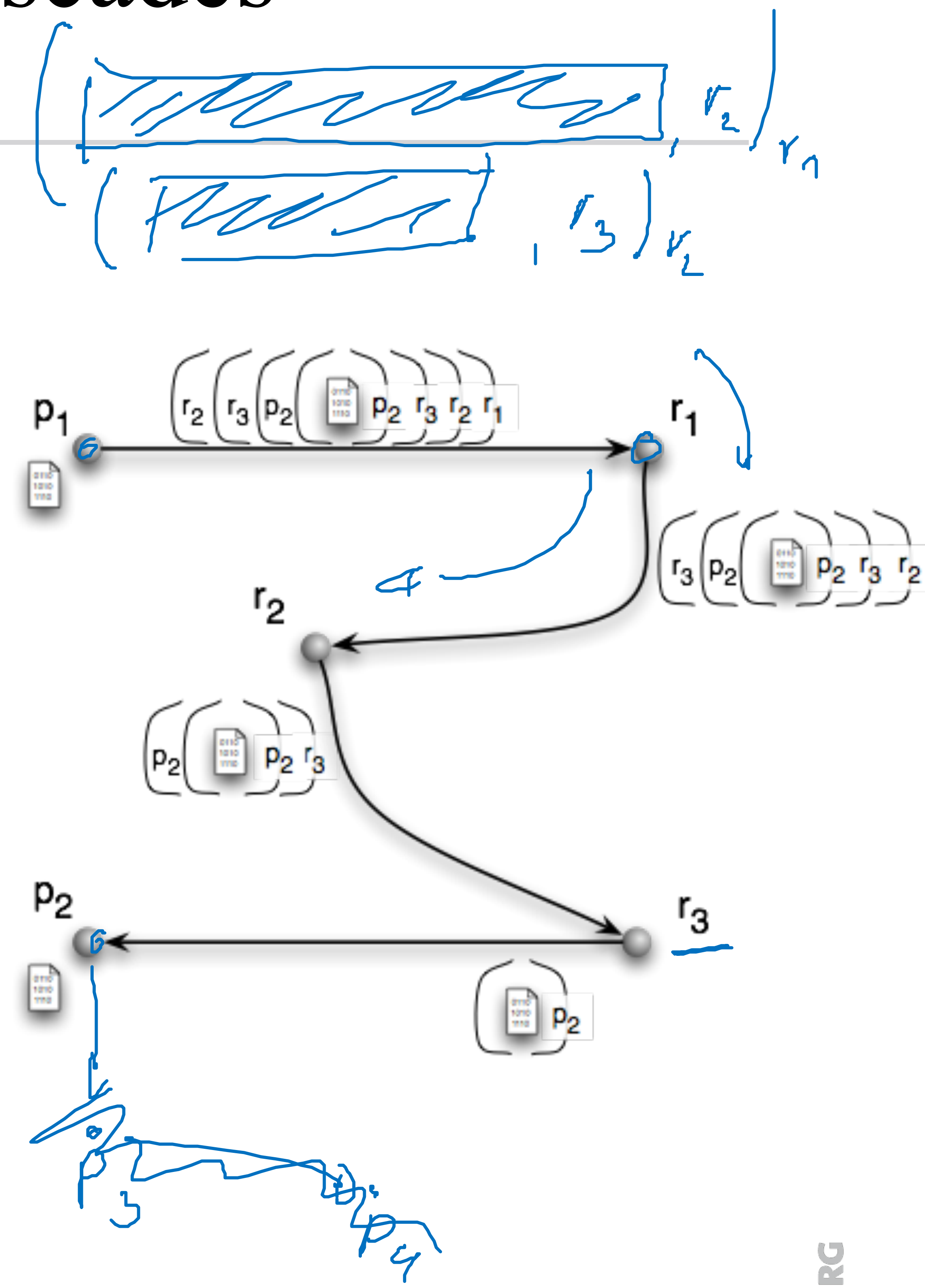
- Encryption function f and decryption function g
  - Known to everybody
- Encryption
  - f(pk,text) = code
  - everybody can generate code

$$(m^s)^p \bmod n = m$$

- Decryption
  - g(sk,code) = code
  - only possibly by receiver

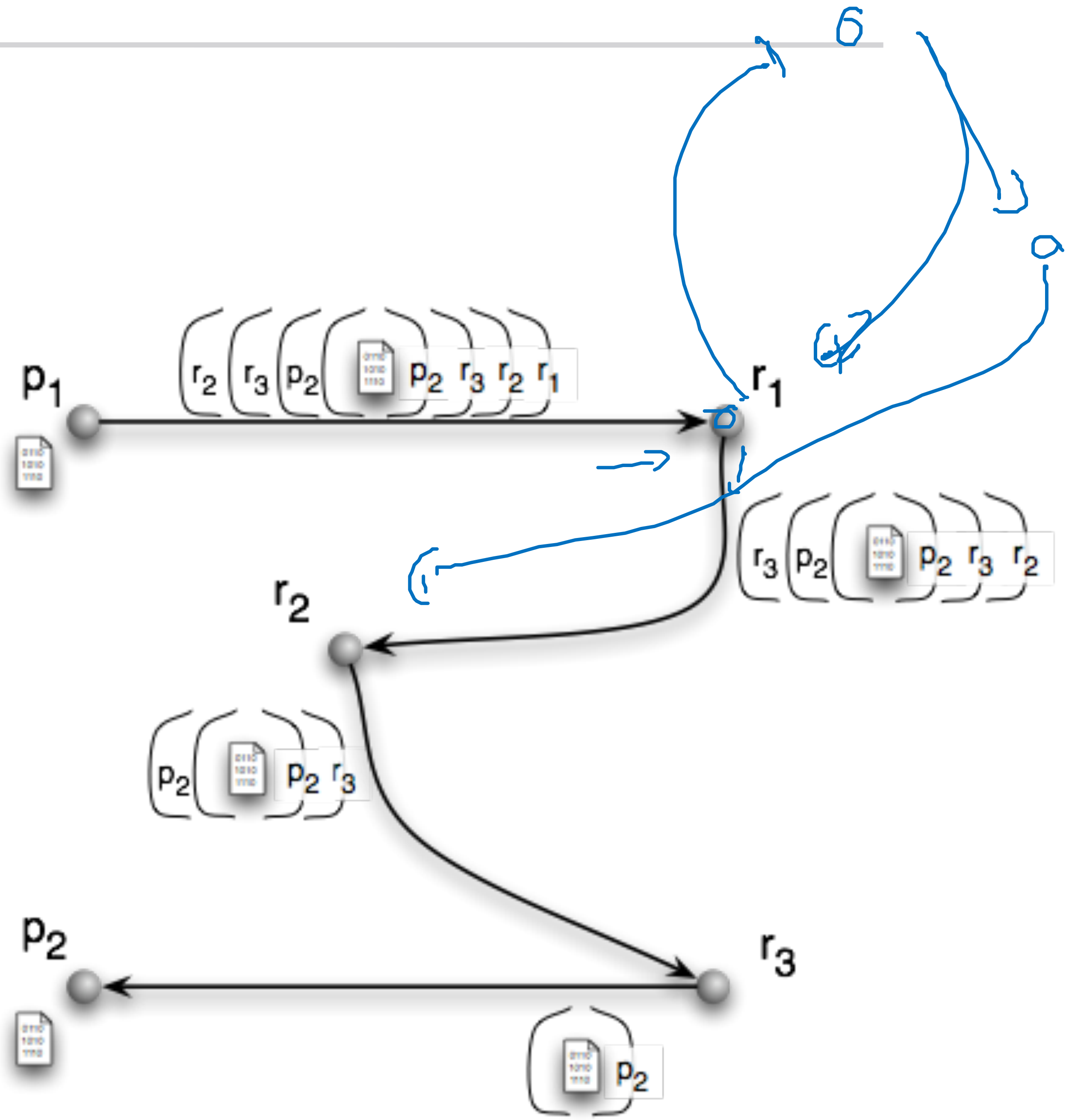
Chaum 's Mix-Cascades

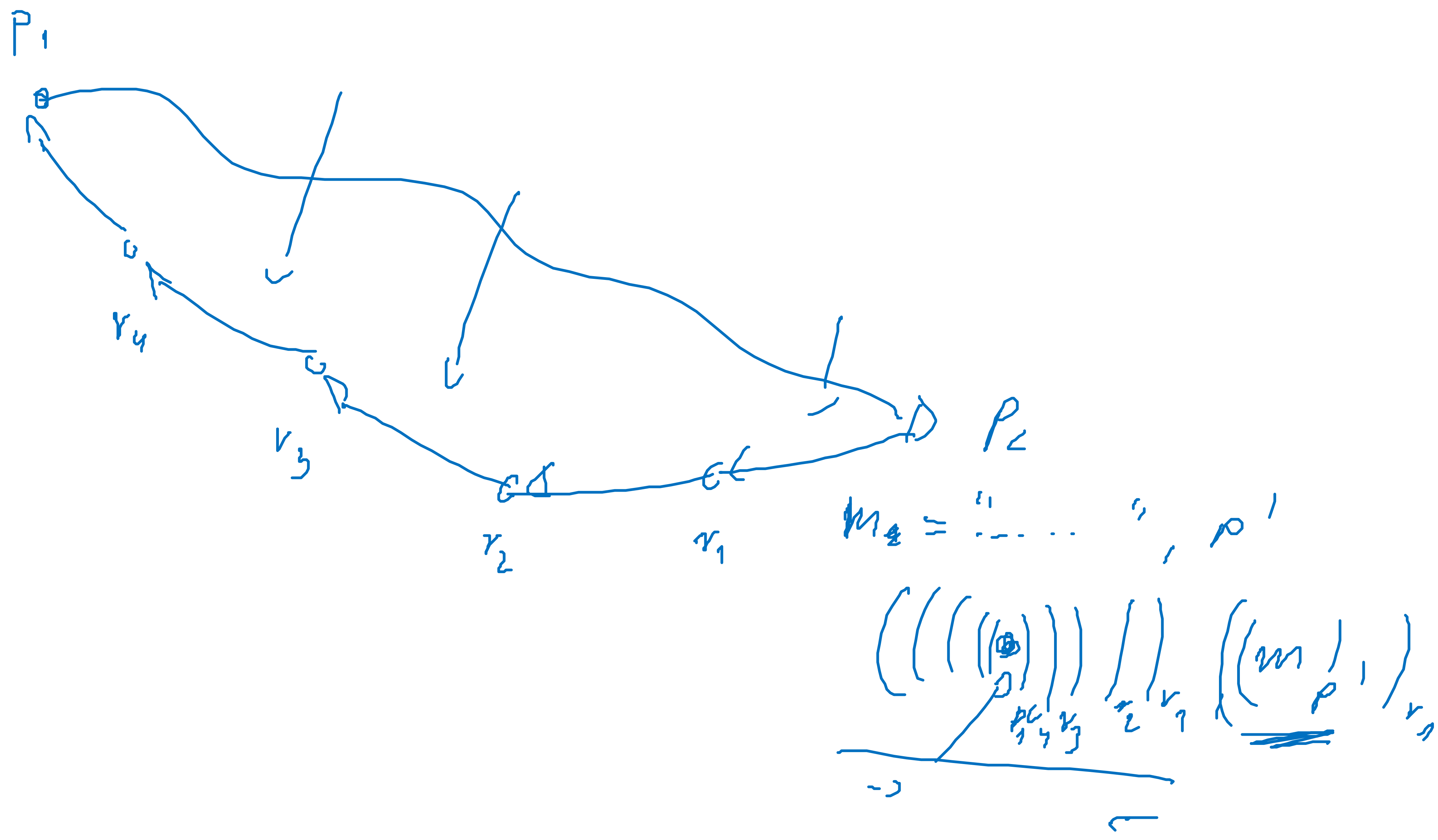
- All peers
  - publish the public keys
  - are known in the network
- The sender p_1 now chooses a route
  - p_1, r_1, r_2, r_3, ..., p_2
- The sender encrypts m according to the public keys from
  - p_2, ... r_3, r_2, r_1
  - and sends the message
  - $f(pk_{k1},(r_2,f(pk_{r2}...f(pk_{rk},(p_2,f(pk_{p2},m)))...)))))$
  - to r_1
- r_1 encrypts the code, deciphers the next hop r_2 and sends it to him
- ...
- until p_2 receives the message and deciphers it



Chaum 's Mix Cascades

- No peer on the route
  - knows its position on the route
  - can decrypt the message
  - knows the final destination
- The receiver does not know the sender
- In addition peers may voluntarily add detour routes to the message
- Chaum 's Mix Cascades
  - aka. Mix Networks or Mixes
  - is safe against all sort of attacks,
  - but not against traffic analysis




P1
P2
v4
v3
v2
v1

TOR - Onion Routers

- David Goldschlag, Michael Reed, and Paul Syverson, 1998
- Goal
  - Preserve private sphere of sender and receiver of a message
  - Safety of the transmitted message
- Prerequisite
  - special infrastructure (Onion Routers)
    - all except some smaller number of exceptions cooperate

TOR - Onion Routers

- Method
  - Mix Cascades (Chaum)
  - Message is sent from source to the target using proxies (Onion Routers)
  - Onion Routers unpredictably choose other routers as intermediate routers
  - Between sender, Onion Routers, and receiver the message is encrypted using symmetric cryptography
  - Every Onion Router only knows the next station
  - The message is encoded like an onion
- TOR is meant as an infrastructure improvement of the Internet
  - not meant as a peer-to-peer network
  - yet, often used from peer-to-peer networks

Other Work based on Onion Routing

- Crowds
  - Reiter & Rubin 1997
  - anonymous web-surfing based on Onion Routers
- Hordes
  - Shields, Levine 2000
  - uses sub-groups to improve Onion Routing
- Tarzan
  - Freedman, 2002
  - A Peer-to-Peer Anonymizing Network Layer
  - uses UDP messages and Chaum Mixes in group to anonymize Internet traffic
  - adds fake traffic against timing attacks

Free-Net

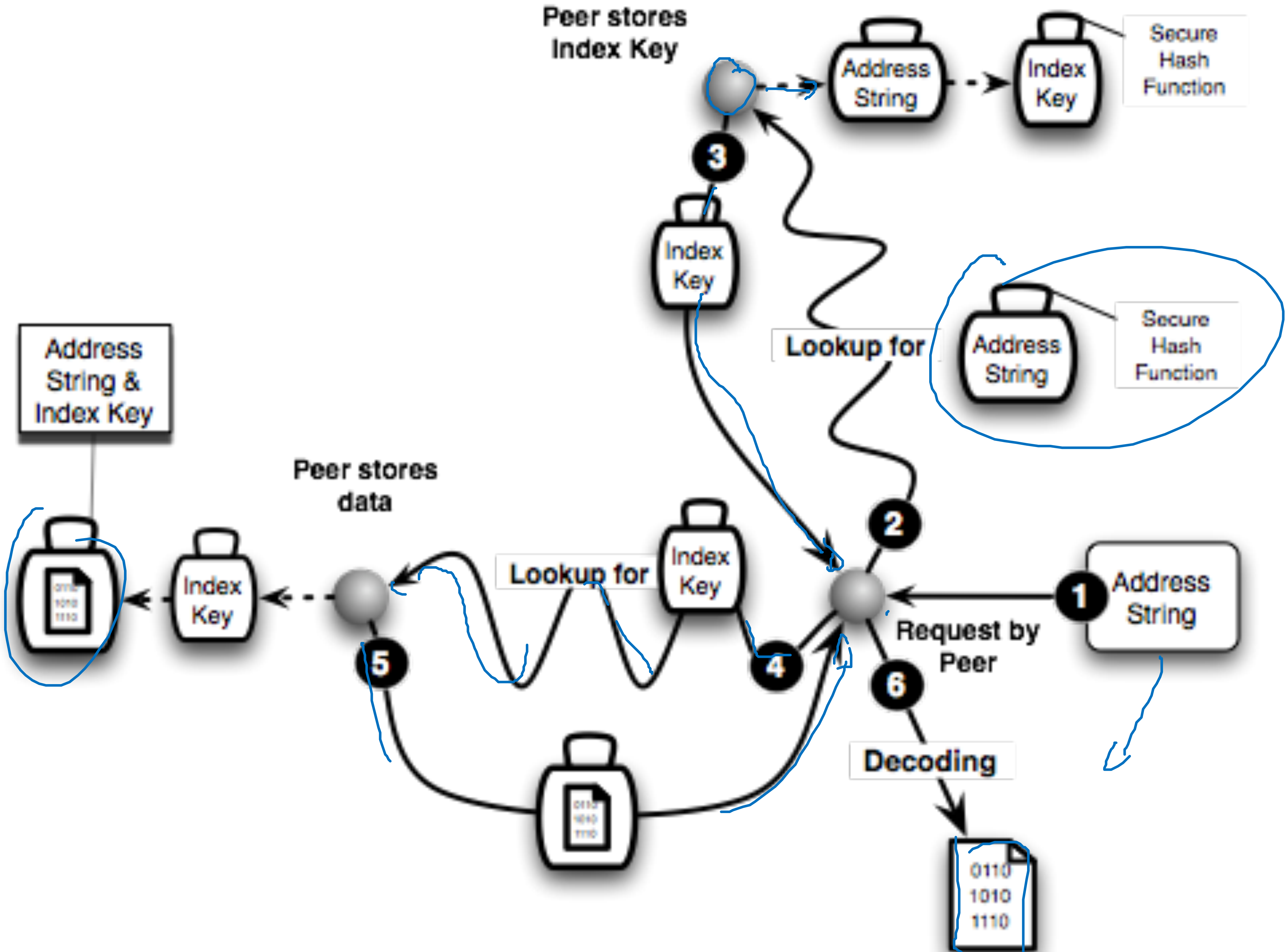
Pseudonym

- Ian Clarke, Oskar Sandberg, Brandon Wiley, Theodore Hong, 2000
- Goal
  - peer-to-peer network
  - allows publication, replication, data lookup
  - anonymity of authors and readers
- Files
  - are encoding location independent
    - by encrypted and pseudonymously signed index files
    - author cannot be identified
  - are secured against unauthorized change or deletion
  - are encoded by keys unknown by the storage peer
    - secret keys are stored elsewhere
  - are replicated
    - on the look up path
  - and erased using “Least Recently Used” (LRU) principle

Free-Net

- Network Structure
  - is similar to Gnutella
  - Free-Net is like Gnutella Pareto distributed
- Storing Files
  - Each file can be found, decoded and read using the encoded address string and the signed subspace key
  - Each file is stored together with the information of the index key but without the encoded address string
  - The storage peer cannot read his files
    - unless he tries out all possible keywords (dictionary attack)
- Storing of index files
  - The address string coded by a cryptographic secure hash function leads to the corresponding peer
    - who stores the index data
      - address string
      - and signed subspace key
  - Using this index file the original file can be found

Free-Net


Peer stores
Index Key
Address
String
Index
Key
Secure
Hash
Function
3
Index
Key
Lookup for
Address
String
Secure
Hash
Function
Address
String &
Index Key
Peer stores
data
2
Lookup for
Index
Key
Index
Key
1
Address
String
5
4
Request by
Peer
6
Decoding
0110
1010
1110

Free-Net

- Lookup
  - steepest-ascent hill-climbing
    - lookup is forwarded to the peer whose ID is closest to the search index
  - with TTL field
    - i.e. hop limit
- Files are moved to new peers
  - when the keyword of the file is similar to the neighbor‘s ID
- New links
  - are created if during a lookup close similarities between peer IDs are discovered

Efficiency of Free-Net

- Network structure of Free-Net is similar to Gnutella
- The lookup time is polynomial on the average

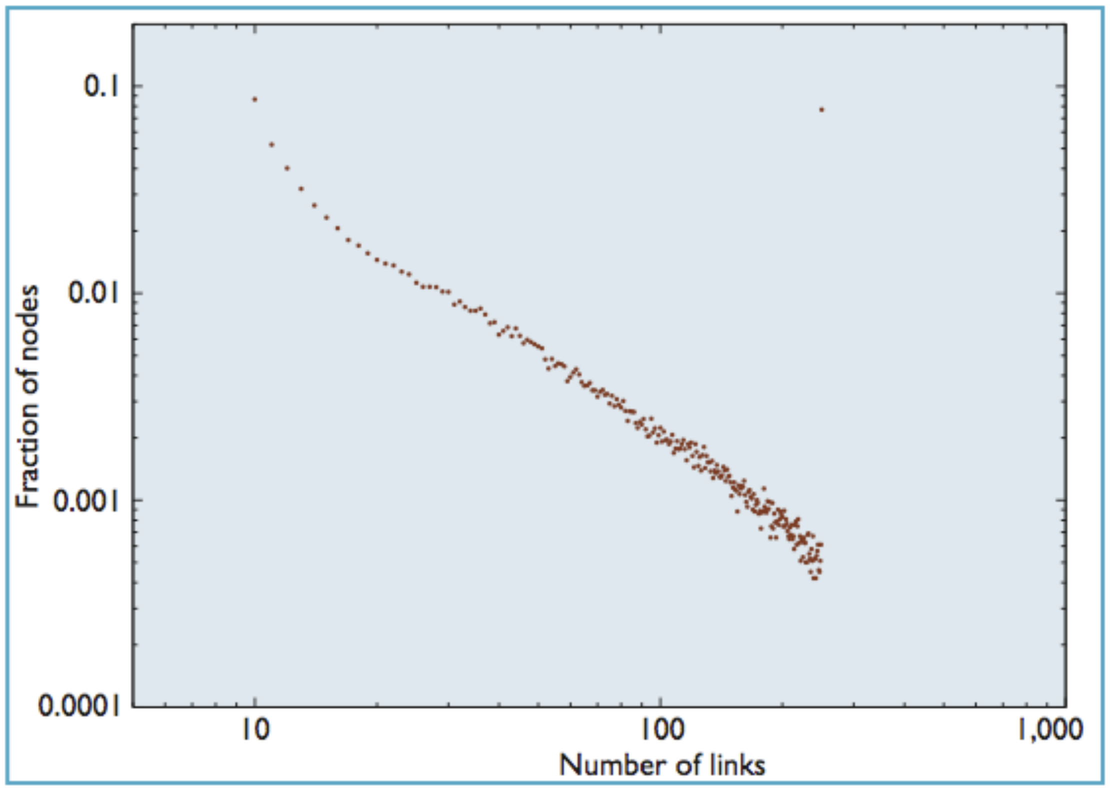


Figure 2. Degree distribution among Freenet nodes. The network shows a close fit to a power-law distribution.

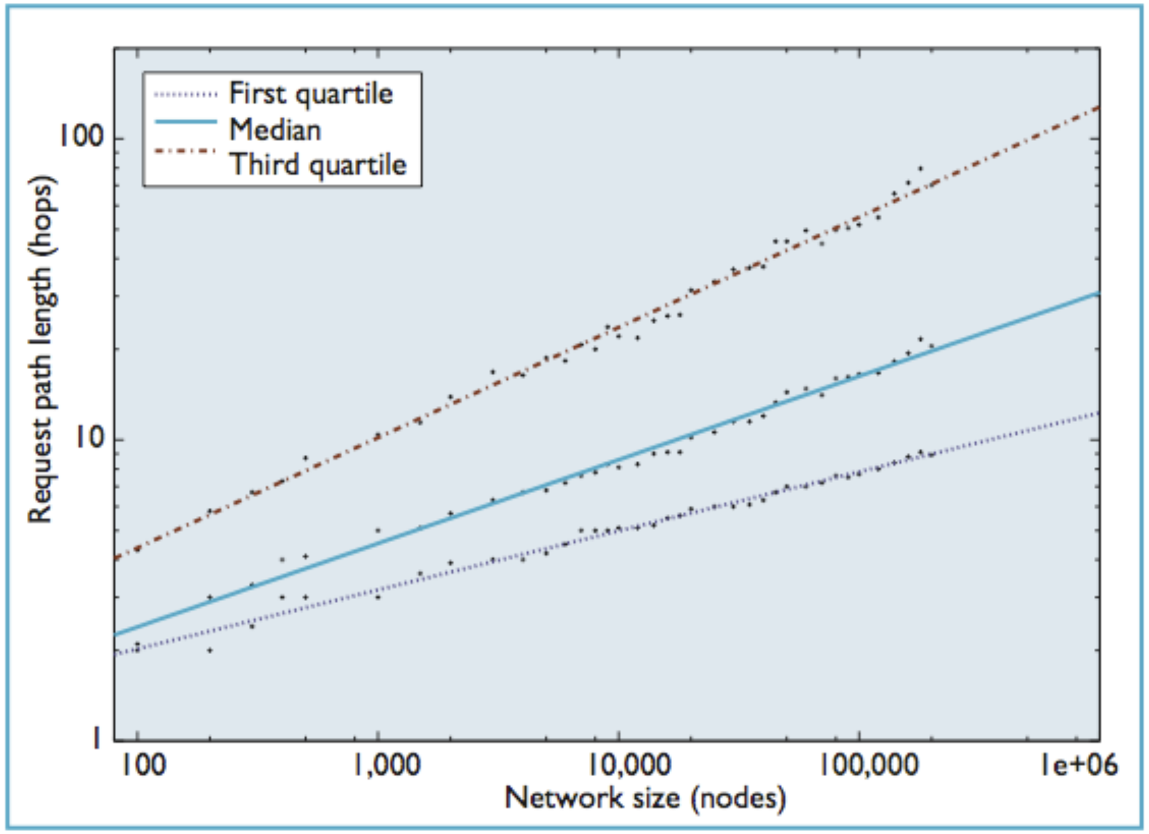


Figure 3. Request path length versus network size. The median path length in the network scales as $N^{0.28}$.

Peer-to-Peer Networks

14 Security

Christian Schindelhauer
Technical Faculty
Computer-Networks and Telematics
University of Freiburg