- Cryptographic Protocols
  - Complexity Theory
    - Language
      - Set of words
    - f is neglible
      - Algorithm is efficient if Runningtime is bounded f(|x|) must be polinomial.
      - Must decrease faster than 1/n^c
      - Example: 2^(-n)
    - noticeable
      - If f is neglible (or poly x neglible) probability is good enough for us
      - Example: n^-2
  - Fiat-Shamir protocol
    - Fiat-Shamir
      - fiat-shamir heuristic
      - Is zero-knowledge
    - Guillou-Quisuatter
      - important! : e requires to be a prime
    - Schnorr
    - Unified
      - Peggy knows x elem G; Vic knows Z elem H
        - fiat-shamir: G=H=Z*_m
        - Guillou-Qusquater: G=H=Z**_m
        - Schnorr: G=Z_q; H s.t. |H|=q
      - c
        - Fiat-Shamir: c={0,1}
        - Guillou-Quisquatter:C={0, ... ,e-1}
        - Schnorr:c={0, ... , q-1}
      - k elem G; t=f(k)
      - F(x)
      - completness
      - soundness
      - zero-knowlwdge
        - c-simulateable
        - Only if poly bounded challenge space
      - proof of knowledge
        - 2-extractable
        - only if a good witness is available
      - Important conditions
        - To calculate T [r] groupOpH z^(-c): c must be efficiently invertable
  - Protocol
    - Hamilton cycle
      - Either reveal permuted cycle
      - Or reveal permutation matrix to original
    - commitment scheme
      - Commitmet schemes
        - Type H
          - GI
        - Type B
          - DL
        - Type H
          - Pedersen
        - Type B
          - QR B
        - Type H
          - QR H
      - commit
        - Peggy generates a blob b=c(Information,randomness)
      - Phases
        - open
      - properties
        - correctness
          - might be called compleness
        - Binding
          - might be perfect binding
            - Type B
            - no exist x1!=x2,r1,r2: c(x1,r1)=c(x2,r2)
            - no trapdoor exists
        - hiding
          - might be perfect hiding (unless already perfect binding)
            - Type H
            - any x1!=x2: c(x1,..) with same distribution c(x2,..)
          - Computational hiding
          - statistical binding
          - trapdoor might exist
        - trapdoor
          - I can make up a blob that I can open in both ways
          - can be made zero knowledge
    - Homomorphic commitment scheme
    - Commitment Transfer Protocol
    - Commitment sharing protocol
  - Adverser
    - passive
    - active
      - publish secret information
        - Is not conidered a problem as honest parties do not conclude anything unless they are malicious
      - withhold information/do not send
      - sending illegal information
  - Proof
    - General
      - Must be concise
      - Must be based on the given problem
      - must proof that the rules to solve it have been followed
    - types
      - static
        - I just send my proof
      - interactive
        - Multiple steps forth and back are required
        - motivation
          - Reduction of transfered information
          - Variety of proofable statements
        - use cases
          - Base for digital signeage
          - Identification protocol
          - multi-party calculations
        - These proofs are in general more powerful
    - criteria
      - Completeness (Vollständigkeit)
        - Accepts all cases where the prover knows the secret
          - usually 100% is required
      - Soundness (Widerspruchsfreiheit)
        - Rejects all cases where the prover does not know the secret
          - Must not be absolute
          - a little probability is acceptable
      - (optional) zero-knowledge
        - trivial: Following the protocol does not reveal any information from the prover to the verifier
        - Formal
          - Zero-Knowledge (ZK)
            - Transscript and simulated Transscript are indistinguishable
            - Runtime is polynomially bounded
          - Black-Box-Zero-Knowledge (BB-ZK)
            - Transscript of an possibly cheating is indstinguishable
            - Runtime is polynomially bounded
          - Honest-verifyer zero-knowledge (HVZK)
        - types
          - perfect
          - statistical
          - computational
      - additional
        - efficient
        - generality
        - Information loss
        - type of security/cryptographic assumptions
    - category
      - Non-Formal
        - statement
        - proof
          - The result
        - verification
          - Steps to verify that the result reflects a valid solution to the statement
      - Formal
        - Class of statements
        - derived statement (eg. a specific case of the class)
        - Proof
        - Verification
    - Content
      - Proof of statement
        - Sudoku has a solution (which I possibly do not know)
      - Proof of knowledge
        - I know a solution to this Sudoku
    - exercices
      - Padlocks
        - knows one out of two
          - Provided solution
            - daisychain padlocks
          - Own solutions
            - Solution 2
              - We have a chemistry box
              - Both locks are in the box
              - Vic cannot see into the box (eppgy can)
              - Peggy may lock a ring with one of the padlocks
            - Solution 1
              - Cover Padlock wheels with Permutation
                - each individually
              - Tell valid permutation
              - downsides
                - not zero knowledge
                - Lock is known afterwards
                - Peggy might just be lucky (not sound)
        - knows one out of 100
          - or
            - interconnect two rings
            - make a ring to a knot
        - knows 2 of 7
          - make two rings
            - not zero-knowledge
          - proof that she can open at least one padlock out of 6
            - see above
      - Kitkat
        - One out of two is distinguishable
        - Three Kitkats are distinguishable
        - One out of three is distinguishable
      - Waldo
        - Know where waldo is
          - cardboard solution
            - Cardboard twice the size with a hole
            - blindly show waldos location
            - Proof later that correct paper has been used by covering the hole and getting the paper
            - not zero knowledge: Leaks information about Waldos surrounding
    - Proof examples
      - Graphs
        - Isomorphismus beweisen
          - Es muss eine Permutationsmatrix geben
        - non-isomorphism beweisen
          - is only non-zero-knowledge to dishonest Verifier
      - RSA
        - Fiat-Shamir
    - Fragen
      - NP
        - Es muss eine Sprache geben, welche durch eine non-det TM akkzeptiert wird
    - Multi-Party Computation
      - Terms
        - Secure
          - If bad guys can not do more in the protocol as in the specification
        - Specification
          - User
          - Truted Party
        - Protocol
          - User
          - Simulating players
            - Simulate trusted party
          - Specifies
            - Input
            - COmpute
            - Output
      - Secret Sharing Schemes
        - A small fraction of the players do not know a value but multiple players might know
        - Formal
          - Correctness
            - After Share: There is a value s' element of F and s'=s if dealer is honest
            - after reconstruct:all players know the value
          - Privacy
            - after Share: all M not Elementt of ?? have no info on S
        - example
          - Shamir Sharing sheme
            - choose a poly function of degree t and (f0)=s
            - give to any party a point on the function F(x) whereas x!=0
              - Any party might then use lagrange interpolation to rebuild f(x0)
                - s=sum(i=1..n, omega_i s_i
                - omega_i=prod(j=1..(i-1),(i+1)..n: -alpha_j/(a_i-alpha_j)