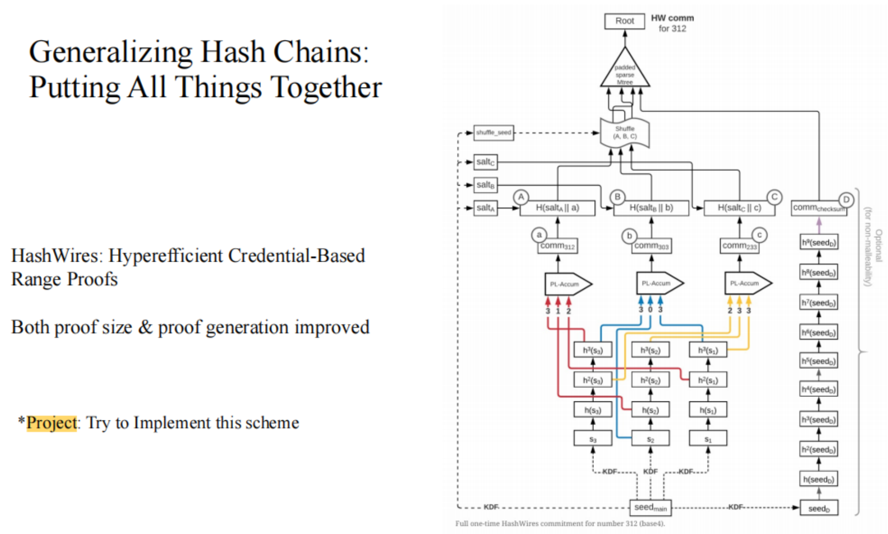
一、SM3

Project1: implement the naïve birthday attack of reduced SM3
Project2: implement the Rho method of reduced SM3
Project3: implement length extension attack for SM3, SHA256, etc
Project4: do your best to optimize SM3 implementation (software)
Project5: Impl Merkle Tree following RFC6962
Project6: Try to Implement this scheme(below)



二、SM4

三、SM2

Project7: report on the application of this deduce technique in Ethereum with ECDSA
Project8: impl sm2 with RFC6979
Project9: verify the above pitfalls with proof-of-concept cod

PART3 Application

3.1 Signatures pitfalls summary

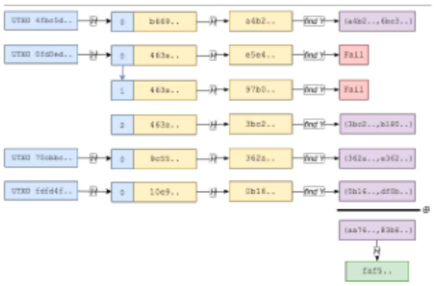
pitfalls	ECDSA	Schnorr	SM2-sig
Leaking k leads to leaking of d	✓	✓	✓
Reusing k leads to leaking of d	✓	✓	✓
Two users, using k leads to leaking of d, that is they can deduce each other's d	✓ RFC 6979	✓ RFC 6979	✓
Malleability, e.g. (r,s) and $(r,-s)$ are both valid signatures, lead to blockchain network split	✓	✓	$r=(e+x_1)\ mod\ n$ $e=Hash(Z_A\|\|M)$
Ambiguity of DER encode could lead to blockchain network split	✓	✓	----
One can forge signature if the verification does not check m	✓	✓	✓
Same d and k with ECDSA, leads to leaking of d	✓	✓	✓

*Project: verify the above pitfalls with proof-of-concept code

Project10: Implement the above ECMH scheme

3.3 UTXO Commitment: Elliptic curve MultiSet Hash

- Homomorphic, or incremental, multiset hash function
  - hash({a}) + hash({b}) = hash({a,b})
- Basic idea: hash each element to an EC point (try and increment)
  - An empty set maps to the infinity point of EC
- Combine/add/remove elements → Points Add of corr. EC Point
- The order of the elements in the multiset does not matter
- Duplicate elements are possible, {a} and {a, a} have different digest
- To update the digest of a multiset, only needs to compute the difference
- Can be constructed on any elliptic curve
- Collision resistant relies on hardness of ECDLP
  - The same security assumption as SM2/ECDSA sign/verify
  - Need more eyes to investigate the security proof of ECMH (maybe worry too much)
- Gains: fast node synchronization, no need to start from the beginning
- Still: you cannot prove to others that you own some Bitcoin efficiently

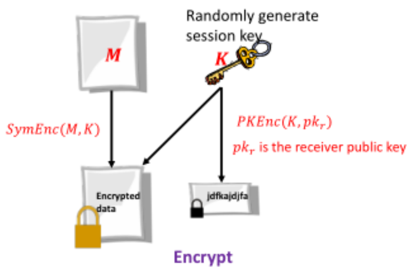


*Project: Implement the above ECMH scheme

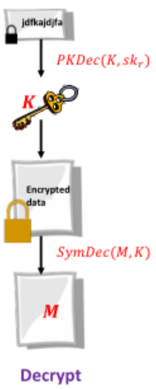
Project11: Implement a PGP scheme with SM2

3.4 PGP

- Generate session key：SM2 key exchange
- Encrypt session key：SM2 encryption
- Encrypt data： Symmetric encryption



Encrypt



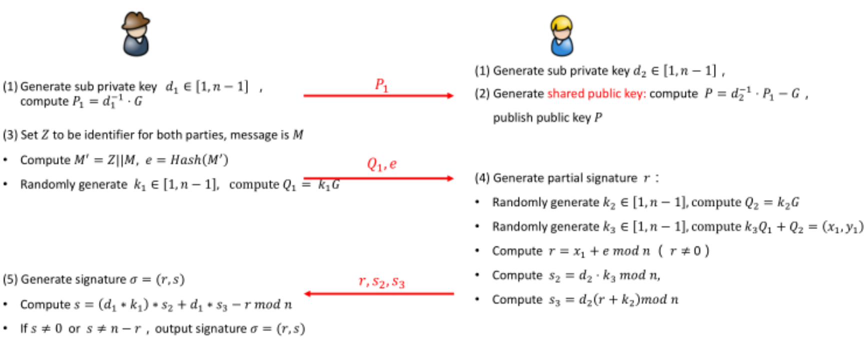
Decrypt

*Project: Implement a PGP scheme with SM2

Project12: implement sm2 2P sign with real network communication

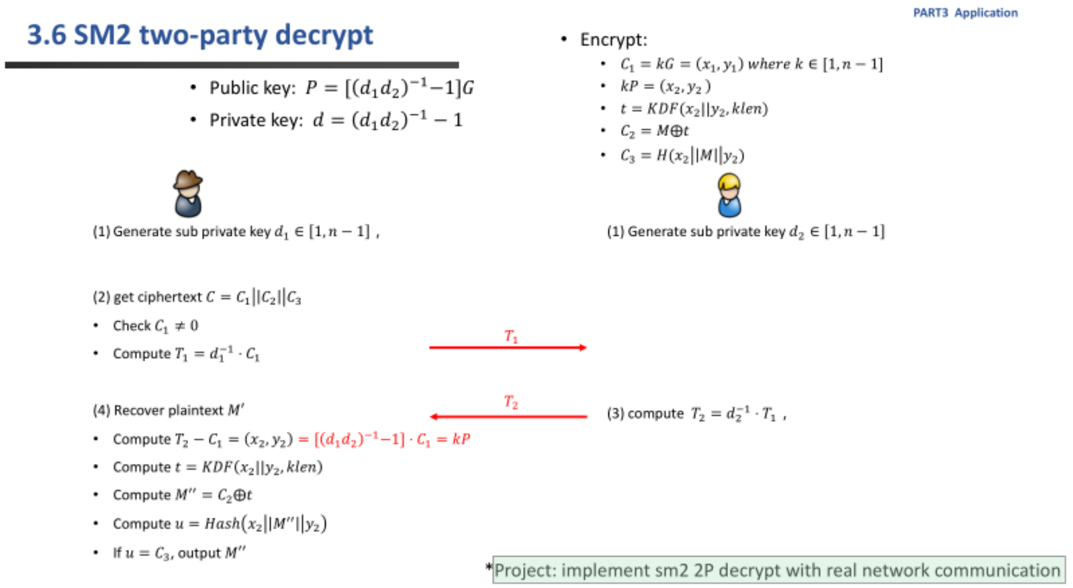
3.5 SM2 two-party sign

- Public key: $P=[(d_1d_2)^{-1}-1]G$
- Private key: $d=(d_1d_2)^{-1}-1$
- Signature
  - $(k_1k_3+k_2)G=(x_1,y_1)$
  - $r=(x_1+e)\ mod\ n$
  - $s=(1+d)^{-1}\cdot\left((k_1k_3+k_2)-r\cdot d\right)\ mod\ n$

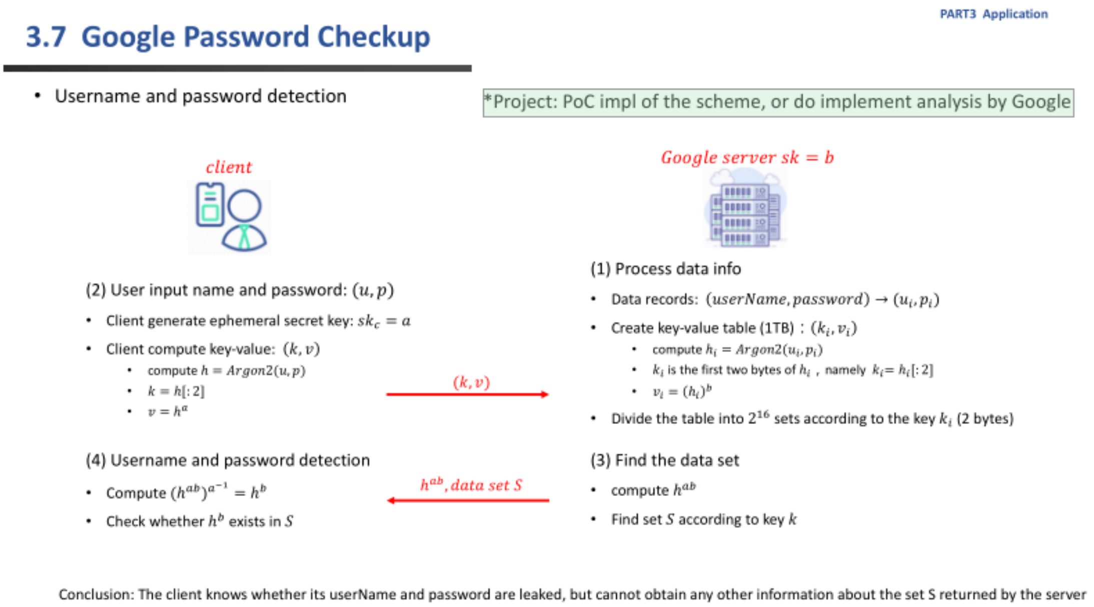


*Project: implement sm2 2P sign with real network communication

Project13: implement sm2 2P decrypt with real network communication



Project14: PoC impl of the scheme, or do implement analysis by Google



四、Bitcon-public

Project15: send a tx on Bitcoin testnet, and parse the tx data down to every bit, better write script yourself

Project16: forge a signature to pretend that you are Satoshi

五、Eth-public

Project17: research report on MPT

六、Real World Cryptanalyses

Project18: Find a key with hash value "*sdu_cst_20220610*" under a message composed of *your name* followed by *your student ID*. For example, "*San Zhan 202000460001*".

Project19: Find a 64-byte message under some k fulfilling that their hash value is symmetrical.

七、zk-SNARKs　project20

Project Idea

1. Write a circuit to prove that your CET6 grade is larger than 425.
   a. Your grade info is like (cn_id, grade, year, sig_by_moe). These grades are published as commitments onchain by MoE.
   b. When you got an interview from an employer, you can prove to them that you have passed the exam without letting them know the exact grade.
2. The commitment scheme used by MoE is SHA256-based.
   a. commit = SHA256(cn_id, grade, year, sig_by_moe, r)