Identity-Based Signature from Lattices without Trapdoors

Pingbin Luo Xinjian Chen Willy Susilo Qiong Huang

lattice signature

lattices hard problem and scheme:

Discrete logarithm problem Given g, g^x find x	→	SIS/LWE problem Given $\boldsymbol{A}, \boldsymbol{As}$ for short $\boldsymbol{s}$, find $\boldsymbol{s}$
Public key encryption ElGamal	→	Public key encryption Regev encrypt
Signature Schnorr RSA Sign	→	Signature Lyubashevsky Signature GPV Signature

lattice signature

The current mainstream lattice signature schemes can be divided into two categories:

"Hash and Sign" in lattice

Falcon [PFHK20]
Mitaka [EPGR22]
HuFu [YJLR23]

core:
preimage sampling [GPV08]
NTRU lattice [DLP14]
Gadget matrix [MP12]

Fiat-Shamir with abort

Dilithium [DLLP18]
qTESLA [ABBK20]
G+G [DPS23]

core:
rejection sampling [Lyu09,Lyu12]
bimodal Gaussians [DDLL13]

lattice signature

Hash and Sign signature scheme:

KeyGen(1^λ):
$(\boldsymbol{A}, \boldsymbol{T}) \leftarrow TrapGen(1^\lambda)$
pk:=$\mathbf{A}$, sk:=$\mathbf{T}$

generate a public matrix and its trapdoor

Sign(sk,m):
$\boldsymbol{s} \leftarrow SamplePre(A, T, \sigma, Hash(m))$
sig:=$\boldsymbol{s}$

generate a SIS instance with the message

Verify(pk,sig,m):
check
$\boldsymbol{As} = Hash(m)$ and $\boldsymbol{s}$ is short

lattice signature

Fiat-Shamir with abort signature scheme:

KeyGen(1^λ):
$\boldsymbol{S} \leftarrow \{-d, \dots, 0, \dots d\}^{m \times k}$
$\boldsymbol{A} \leftarrow Z_q^{n \times m}$
$\boldsymbol{T} := \boldsymbol{AS}$
$pk := (\boldsymbol{A}, \boldsymbol{T}), sk := \boldsymbol{S}$

generate k SIS instances

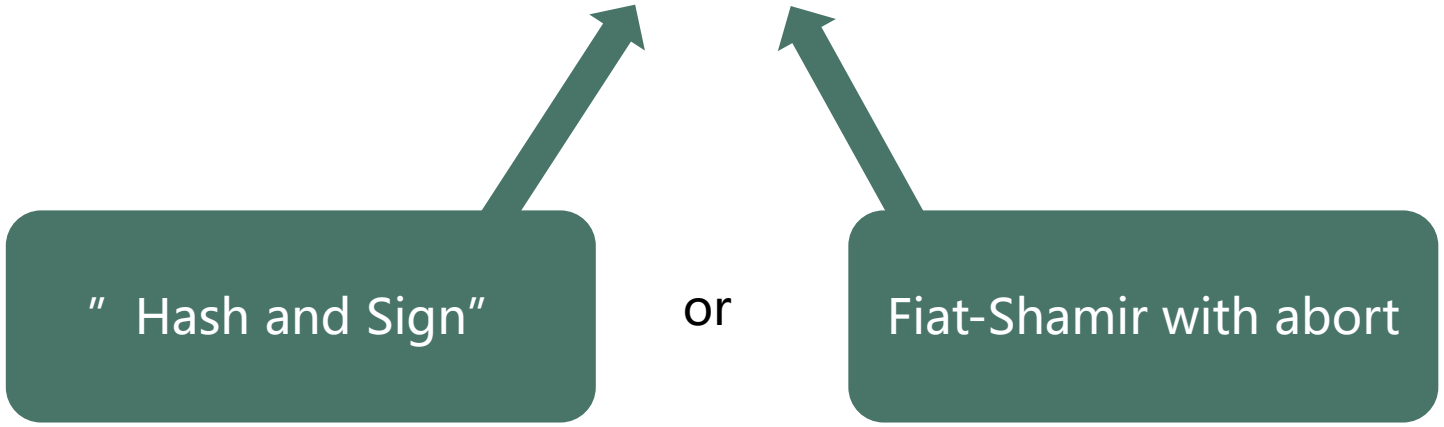
Sign(sk,m):
$\boldsymbol{y} \leftarrow D_\sigma^m$
$\boldsymbol{c} := H(\boldsymbol{Ay}, m)$
$\boldsymbol{z} := \boldsymbol{y} + \boldsymbol{Sc}$
if $\boldsymbol{z} \notin D_\sigma^m$, restart
sig:= $(\boldsymbol{z}, \boldsymbol{c})$

Verify(pk,sig,m):
check
$\boldsymbol{z}$ is short and $\boldsymbol{c} = H(\boldsymbol{Az} - \boldsymbol{Tc}, \mu)$

identity base signature

An identity-based signature scheme consists of four algorithms:

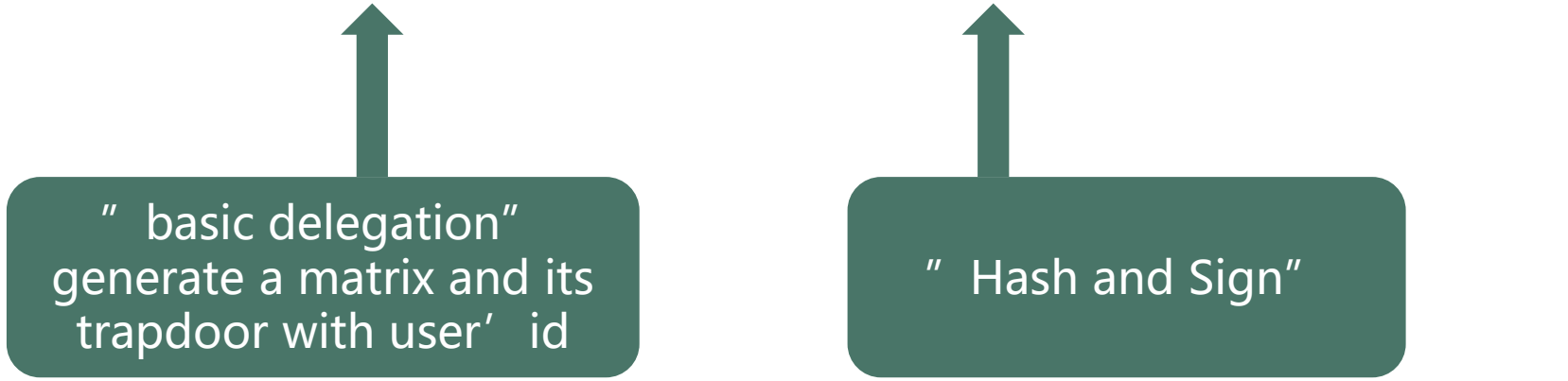
Setup Extract Sign Verify



identity base signature

An identity-based signature scheme consists of four algorithms:

Setup Extract Sign Verify



advantage: proved tight secure in QROM [PW21]

disadvantage: large public key and signature size

identity base signature

An identity-based signature scheme consists of four algorithms:

Setup Extract Sign Verify

this approch is actually a kind of “Hash and Sign” signature!

advantage: easy implement [TH14]

disadvantage: all the disadvantage of “Hash and Sign” signature

lattice trapdoors

with a lattice’ s trapdoor, one can generate a SIS or LWE instance in polynomial time

lattice trapdoors

Gadget matrix based trapdoor [MP12]:
large public key and signature size,
slow running time

NTRU based trapdoor [DLP14]:
fast, small signature size
but may be not secure in large modulus
no basic delegation

Our contributions

A lattice based IBS without using lattice trapdoors

Setup Extract Sign Verify

generate a SIS instance with user’ id

without using trapdoors

Fiat-Shamir with abort

easy implement, small matrix size and signature size, fast

key generation

How to generate a private key

Setup(1^λ):
$\boldsymbol{A} \leftarrow Z_q^{n\times m}$
$\boldsymbol{S} \leftarrow Z_\beta^m$
$\boldsymbol{T} := \boldsymbol{AS}$
mpk:=(**A**,**T**), msk:=**S**

KeyGen(msk,id):
$\boldsymbol{y} \leftarrow D_\sigma^m$
$\boldsymbol{w} \leftarrow \boldsymbol{Ay}$
$\boldsymbol{c} := Hash(\boldsymbol{w}, id)$
$\boldsymbol{z} := \boldsymbol{y} + \boldsymbol{Sc}$
if $\boldsymbol{z} \notin D_\sigma^m$, restart
pk:=$\boldsymbol{w}$, sk:=$\boldsymbol{z}$ ⟵ $\boldsymbol{Az} = \boldsymbol{w} + \boldsymbol{T} \cdot Hash(\boldsymbol{w}, id)$ and $\boldsymbol{z}$ is short

generate a SIS instance

lattice signature

Setup(1^λ):
$\boldsymbol{A} \leftarrow Z_q^{n\times m}$
$\boldsymbol{S} \leftarrow Z_\beta^{m\times k}$
$\boldsymbol{T} := \boldsymbol{AS}$
mpk:=(**A**,**T**), msk:=**S**

KeyGen(msk,id):
$\boldsymbol{Y} \leftarrow D_\sigma^{m\times k}$
$\boldsymbol{W} \leftarrow \boldsymbol{AY}$
$\boldsymbol{C} := Hash(\boldsymbol{W}, id)$
$\boldsymbol{Z} := \boldsymbol{Y} + \boldsymbol{SC}$
if $\boldsymbol{Z} \notin D_\sigma^{m\times k}$, restart
sk:=($\boldsymbol{Z}$,$\boldsymbol{W}$)

Sign(pk,sk,id,μ):
$\boldsymbol{y} \leftarrow D_\rho^m$
$\boldsymbol{w} \leftarrow \boldsymbol{Ay}$
$\boldsymbol{c} := Hash(\boldsymbol{w}, id, \mu)$
$\boldsymbol{z} := \boldsymbol{y} + \boldsymbol{Zc}$
if $\boldsymbol{z} \notin D_\rho^m$, restart
sig:=($\boldsymbol{z}$,$\boldsymbol{c}$,$\boldsymbol{W}$)

Verify(mpk,id,sig,μ):
$\boldsymbol{T}' := \mathbf{W} + \boldsymbol{T} \cdot Hash(\boldsymbol{W}, id)$
check
$\boldsymbol{z}$ is short
$\boldsymbol{c} = Hash(\boldsymbol{Az} - \boldsymbol{T}'\boldsymbol{c}, id, \mu)$

Fiat-Shamir with abort

It just a framwork, with latest optimization techniques, it could do better

modules lattice [LS15], signature compression [DLLP18], convolved Gaussians [DPS23]

comparison

scheme	hard problem	signature size	setup	extract	sign	verify
FL23	SIS	>5MB	-	-	-	-
SPMC23	Ring-SIS	>1MB	-	-	-	-
XHGG16	NTRU,Ring-SIS	12918 Bytes	11652ms	70ms	23ms	12ms
ours	MLWE,MSIS	6691 Bytes	9ms	47ms	28ms	11ms

Thank you!