Threshold Cryptography based on Class Groups of Imaginary Quadratic Fields

Guilhem Castagnos

Journées Nationales 2025 du GDR Sécurité Informatique

Caen, Tuesday, June 24, 2025

université de BORDEAUX

Outline

Traditional PKE

- Bob : $(pk, sk) \leftarrow \mathsf{KeyGen}(1^k)$
- Publish pk

Traditional PKE

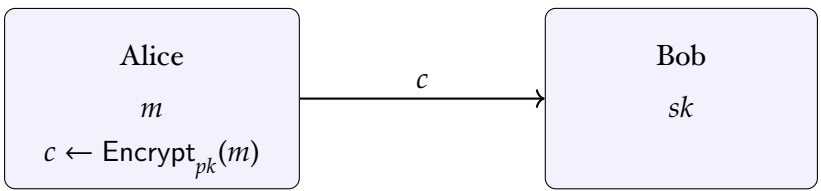
- Bob : $(pk, sk) \leftarrow \mathsf{KeyGen}(1^k)$
- Publish pk

Traditional PKE

- Bob : $(pk, sk) \leftarrow \mathsf{KeyGen}(1^k)$
- Publish pk

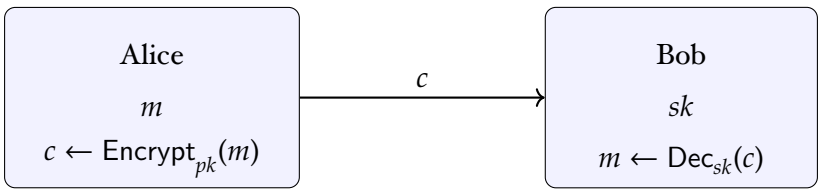
Traditional PKE

- Bob : $(pk, sk) \leftarrow \mathsf{KeyGen}(1^k)$
- Publish pk



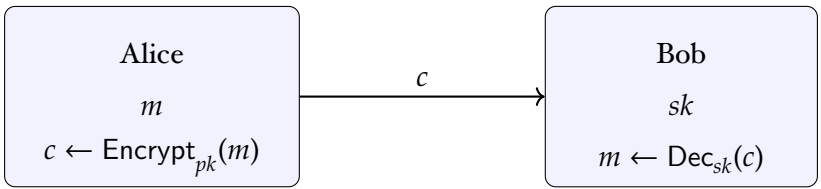
Traditional PKE

- Bob : $(pk, sk) \leftarrow \mathsf{KeyGen}(1^k)$
- Publish pk



Traditional PKE

- Bob : $(pk, sk) \leftarrow \mathsf{KeyGen}(1^k)$
- Publish pk



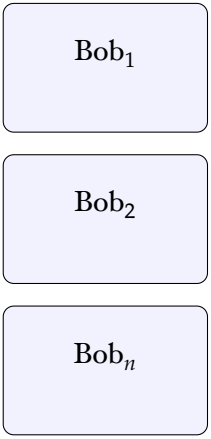
- Linearly Homomorphic Encryption (LHE):

$$\mathsf{Encrypt}_{pk}(m_1) \otimes \mathsf{Encrypt}_{pk}(m_2) \rightsquigarrow \mathsf{Encrypt}_{pk}(m_1 + m_2)$$

$$\mathsf{Encrypt}_{pk}(m)^{\otimes a} \rightsquigarrow \mathsf{Encrypt}_{pk}(a \cdot m)$$

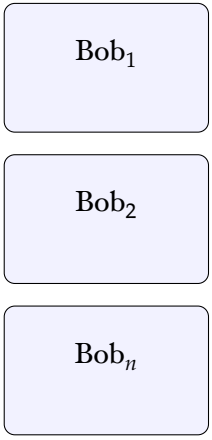
Threshold LHE

- n Bobs collaborate to decrypt a ciphertext



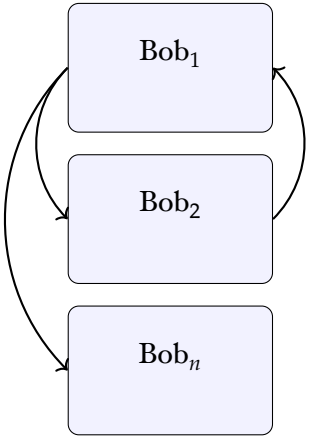
Threshold LHE

- n Bobs collaborate to decrypt a ciphertext
- Distributed Key Generation (DKG)



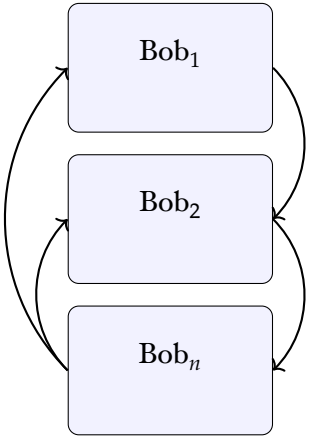
Threshold LHE

- n Bobs collaborate to decrypt a ciphertext
- Distributed Key Generation (DKG)



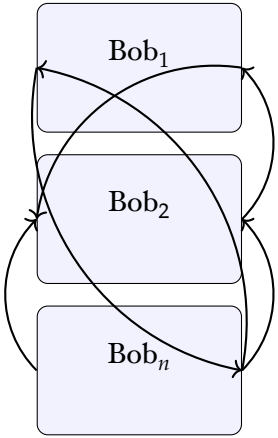
Threshold LHE

- n Bobs collaborate to decrypt a ciphertext
- Distributed Key Generation (DKG)



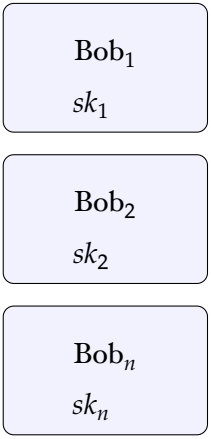
Threshold LHE

- n Bobs collaborate to decrypt a ciphertext
- Distributed Key Generation (DKG)



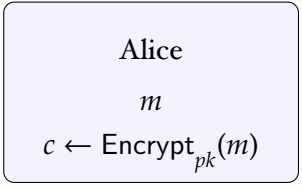
Threshold LHE

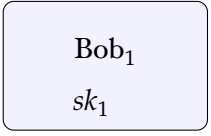
- n Bobs collaborate to decrypt a ciphertext
- Distributed Key Generation (DKG) $\rightsquigarrow pk$

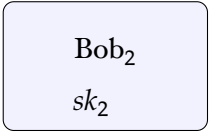


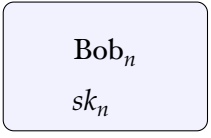
Threshold LHE

- n Bobs collaborate to decrypt a ciphertext



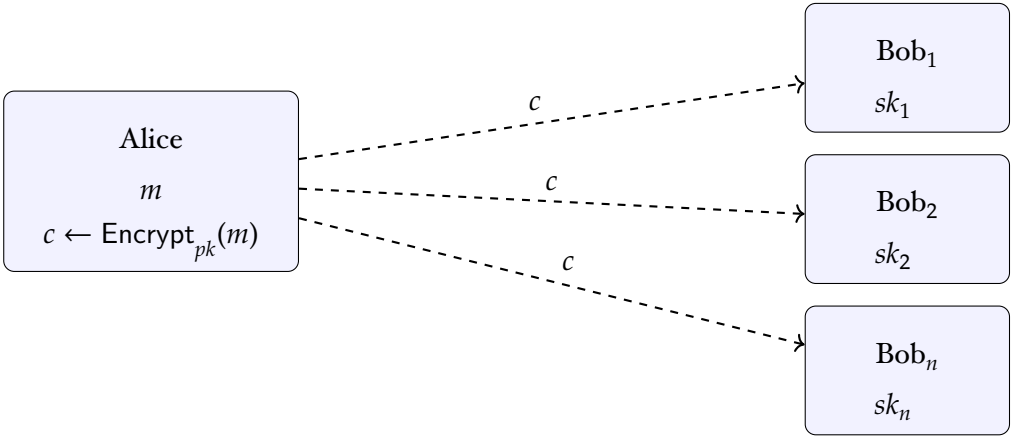






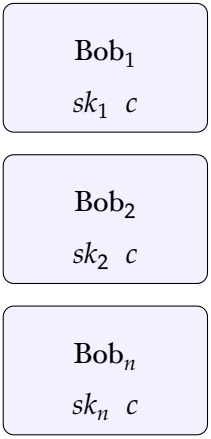
Threshold LHE

- n Bobs collaborate to decrypt a ciphertext



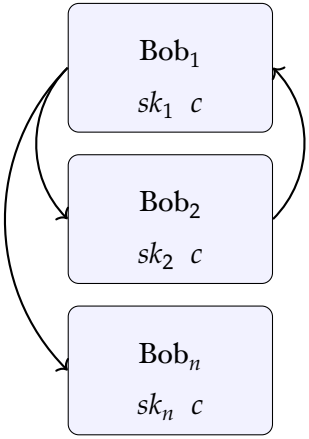
Threshold LHE

- n Bobs collaborate to decrypt a ciphertext
- Distributed Decryption



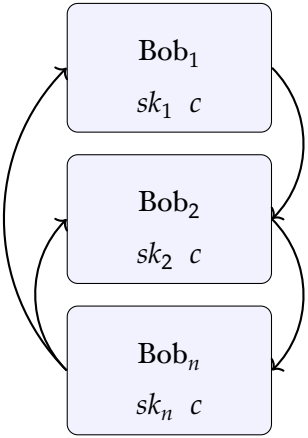
Threshold LHE

- n Bobs collaborate to decrypt a ciphertext
- Distributed Decryption



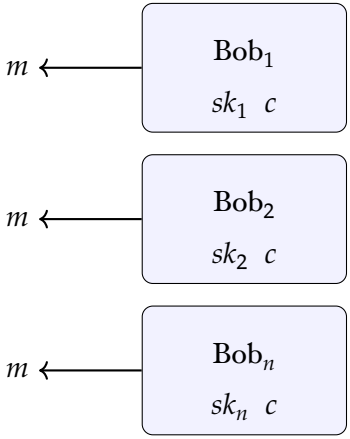
Threshold LHE

- n Bobs collaborate to decrypt a ciphertext
- Distributed Decryption



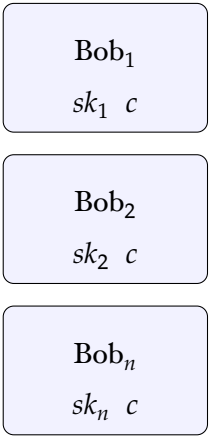
Threshold LHE

- n Bobs collaborate to decrypt a ciphertext
- Distributed Decryption

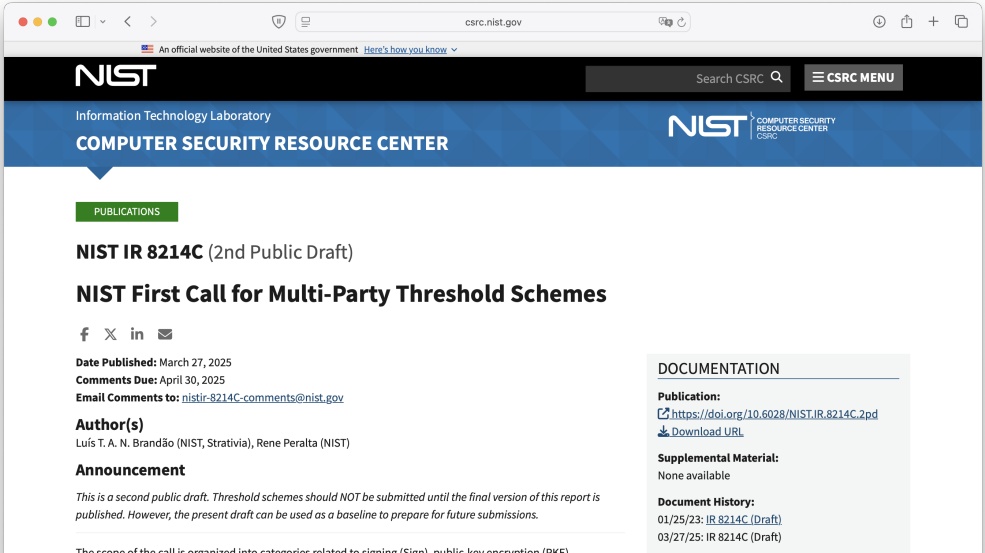


Threshold LHE

- n Bobs collaborate to decrypt a ciphertext
- Threshold:
  - $t < n$ Bobs $\rightsquigarrow \emptyset$ info on m
  - $t + 1$ Bobs $\rightsquigarrow m$
  - Assume $t < n/2$ to get guaranteed output delivery



NIST Call


csrc.nist.gov
An official website of the United States government Here's how you know
NIST
Search CSRC
CSRC MENU
Information Technology Laboratory
COMPUTER SECURITY RESOURCE CENTER
NIST COMPUTER SECURITY RESOURCE CENTER CSRC
PUBLICATIONS
NIST IR 8214C (2nd Public Draft)
NIST First Call for Multi-Party Threshold Schemes
Date Published: March 27, 2025
Comments Due: April 30, 2025
Email Comments to: nistir-8214C-comments@nist.gov
Author(s)
Luís T. A. N. Brandão (NIST, Strativia), Rene Peralta (NIST)
Announcement
This is a second public draft. Threshold schemes should NOT be submitted until the final version of this report is published. However, the present draft can be used as a baseline to prepare for future submissions.
DOCUMENTATION
Publication:
https://doi.org/10.6028/NIST.IR.8214C.2pd
Download URL
Supplemental Material:
None available
Document History:
01/25/23: IR 8214C (Draft)
03/27/25: IR 8214C (Draft)

Applications of TLHE

Electronic Voting Scheme: Yes/No choice: vote 1 or 0

$$\left.\begin{array}{lcll} \text{Alice}_1: & 0 & \to & c_1 := \text{Encrypt}_{pk}(0) \\ \text{Alice}_2: & 1 & \to & c_2 := \text{Encrypt}_{pk}(1) \\ \vdots & \vdots & \vdots & \vdots \\ \text{Alice}_\ell: & 1 & \to & c_\ell := \text{Encrypt}_{pk}(1) \end{array}\right\} \rightsquigarrow c := \otimes_{i=1}^{\ell} c_i$$

Applications of TLHE

Electronic Voting Scheme: Yes/No choice: vote 1 or 0

$$\left.\begin{array}{lcll} \text{Alice}_1: & 0 & \to & c_1 := \text{Encrypt}_{pk}(0) \\ \text{Alice}_2: & 1 & \to & c_2 := \text{Encrypt}_{pk}(1) \\ \vdots & \vdots & \vdots & \vdots \\ \text{Alice}_\ell: & 1 & \to & c_\ell := \text{Encrypt}_{pk}(1) \end{array}\right\} \rightsquigarrow c := \otimes_{i=1}^{\ell} c_i$$

Bob

sk

Applications of TLHE

Electronic Voting Scheme: Yes/No choice: vote 1 or 0

$$\left.\begin{array}{lccl} \text{Alice}_1: & 0 & \to & c_1 := \text{Encrypt}_{pk}(0) \\ \text{Alice}_2: & 1 & \to & c_2 := \text{Encrypt}_{pk}(1) \\ \vdots & \vdots & \vdots & \vdots \\ \text{Alice}_\ell: & 1 & \to & c_\ell := \text{Encrypt}_{pk}(1) \end{array}\right\} \rightsquigarrow c := \otimes_{i=1}^{\ell} c_i$$

Bob

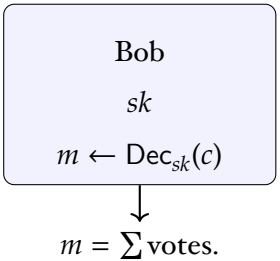
sk

$m \leftarrow \text{Dec}_{sk}(c)$

Applications of TLHE

Electronic Voting Scheme: Yes/No choice: vote 1 or 0

$$\left.\begin{array}{lccl} \text{Alice}_1: & 0 & \to & c_1 := \text{Encrypt}_{pk}(0) \\ \text{Alice}_2: & 1 & \to & c_2 := \text{Encrypt}_{pk}(1) \\ \vdots & \vdots & \vdots & \vdots \\ \text{Alice}_\ell: & 1 & \to & c_\ell := \text{Encrypt}_{pk}(1) \end{array}\right\} \rightsquigarrow c := \otimes_{i=1}^{\ell} c_i$$



Applications of TLHE

Electronic Voting Scheme: Yes/No choice: vote 1 or 0

$$\left.\begin{array}{lccl} \text{Alice}_1: & 0 & \to & c_1 := \text{Encrypt}_{pk}(0) \\ \text{Alice}_2: & 1 & \to & c_2 := \text{Encrypt}_{pk}(1) \\ \vdots & \vdots & \vdots & \vdots \\ \text{Alice}_\ell: & 1 & \to & c_\ell := \text{Encrypt}_{pk}(1) \end{array}\right\} \rightsquigarrow c := \otimes_{i=1}^{\ell} c_i$$

Bob

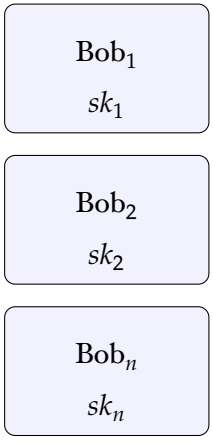
sk

$1 \leftarrow \text{Dec}_{sk}(c_2)$

Applications of TLHE

Electronic Voting Scheme: Yes/No choice: vote 1 or 0

$$\left.\begin{array}{lcll} \text{Alice}_1: & 0 & \to & c_1 := \text{Encrypt}_{pk}(0) \\ \text{Alice}_2: & 1 & \to & c_2 := \text{Encrypt}_{pk}(1) \\ \vdots & \vdots & \vdots & \vdots \\ \text{Alice}_\ell: & 1 & \to & c_\ell := \text{Encrypt}_{pk}(1) \end{array}\right\} \rightsquigarrow c := \otimes_{i=1}^{\ell} c_i$$



Applications of TLHE

Multi-Party Computation: Cramer, Damgård, Nielsen, EC'01

computation of an arithmetic function over a ring $\mathscr{R}$

- Parties share a decryption key for a TLHE over $\mathscr{R}$
- Encrypt their secret input and broadcast $\mathsf{Encrypt}_{pk}(x_i)$
- Without interaction: Addition of secret values, multiplication by a public value

$$\mathsf{Encrypt}_{pk}(x) \otimes \mathsf{Encrypt}_{pk}(y) \; ; \; \mathsf{Encrypt}_{pk}(x)^{\otimes a}$$

- Multi-Party subprotocol to compute

$$\mathsf{Encrypt}_{pk}(xy) \text{ from } \mathsf{Encrypt}_{pk}(x) \text{ and } \mathsf{Encrypt}_{pk}(y)$$

- Final threshold decryption to compute the result

Applications of TLHE

Multi-Party Computation

- CDN less efficient than MPC with pre-processing (SPDZ)
- YOSO! Gentry, Halevi, Krawczyk, Magri, Nielsen, Rabin, Yakoubov, C'21
- Large universe of M servers
- Computation done by a committee of $N \ll M$ servers
- Goal: Hide the committee to the adversary
- You Only Speak Once: committee sends only one round of messages
- Instantiation from CDN

Examples of TLHE

ElGamal in the Exponent (C'84)

- $sk = x \quad pk = g^x \quad c = (g^r, g^m pk^r)$
- Discrete Log. computation during decryption: $g^m \rightsquigarrow m$
- Limited homomorphism (few $c \otimes c'$, no $c^{\otimes a}$ for large a)
- Sufficient for voting (Helios, Belenios)

Paillier (EC'99)

- $sk = p, q \quad pk = \mathrm{N} = pq \quad c = (1+\mathrm{N})^m r^{\mathrm{N}} \pmod{\mathrm{N}^2}$
- Easy Discrete Log. computation during decryption
- Linear homomorphism without restriction
- Requires a trusted setup for Keygen, or costly MPC, $\mathscr{R} = \mathbf{Z}/\mathrm{N}\mathbf{Z}$

Examples of TLHE

ElGamal in the Exponent (C'84)

- $sk = x \quad pk = g^x \quad c = (g^r, g^m pk^r)$
- Discrete Log. computation during decryption: $g^m \rightsquigarrow m$
- Limited homomorphism (few $c \otimes c'$, no $c^{\otimes a}$ for large a)
- Sufficient for voting (Helios, Belenios)

Paillier (EC'99)

- $sk = p, q \quad pk = \mathrm{N} = pq \quad c = (1+\mathrm{N})^m r^{\mathrm{N}} \pmod{\mathrm{N}^2}$
- Easy Discrete Log. computation during decryption
- Linear homomorphism without restriction
- Requires a trusted setup for Keygen, or costly MPC, $\mathscr{R} = \mathbf{Z}/\mathrm{N}\mathbf{Z}$

What about CL ?

Outline

Defining Class Groups in one slide

Definition

- Fundamental Discriminant $\Delta_K < 0$ square free, $\Delta_\ell = \ell^2 \Delta_K$

$$\mathcal{O}_{\Delta_\ell} = \mathbf{Z} + \frac{\Delta_\ell + \sqrt{\Delta_\ell}}{2}\mathbf{Z}$$

- Class group of $\mathcal{O}_{\Delta_\ell}$

$$C(\mathcal{O}_{\Delta_\ell}) := I(\mathcal{O}_{\Delta_\ell})/P(\mathcal{O}_{\Delta_\ell})$$

Properties

- Description of the group: Δ_ℓ
- Equivalence relation: $\mathfrak{a} \sim \mathfrak{b} \iff \exists\, \alpha \in K^*,\ \mathfrak{b} = \alpha\,\mathfrak{a}$
- Short representations, efficient computation of the group law
- Class Number: $h(\mathcal{O}_{\Delta_\ell}) \approx \sqrt{|\Delta_\ell|}$

A tiny class group

- $\Delta_K = -131$
- $\mathscr{C}(\mathscr{O}_{\Delta_K})$ has 5 classes. Reduced elements:

$$\left[3\mathbf{Z} + \frac{-1+\sqrt{\Delta_K}}{2}\mathbf{Z}\right] \quad \left[5\mathbf{Z} + \frac{3+\sqrt{\Delta_K}}{2}\mathbf{Z}\right] \quad \left[5\mathbf{Z} + \frac{-3+\sqrt{\Delta_K}}{2}\mathbf{Z}\right] \quad \left[3\mathbf{Z} + \frac{1+\sqrt{\Delta_K}}{2}\mathbf{Z}\right]$$

$$\left[\mathbf{Z} + \frac{\Delta_K+\sqrt{\Delta_K}}{2}\mathbf{Z}\right]$$

- A multiplication :

$$\left(3\mathbf{Z} + \frac{-1+\sqrt{\Delta_K}}{2}\mathbf{Z}\right)\left(5\mathbf{Z} + \frac{3+\sqrt{\Delta_K}}{2}\mathbf{Z}\right) = \left(15\mathbf{Z} + \frac{23+\sqrt{\Delta_K}}{2}\mathbf{Z}\right)$$

- Reduction : $3\mathbf{Z} + \frac{1+\sqrt{\Delta_K}}{2}\mathbf{Z}$ (by multiplication by $\frac{-7-\sqrt{\Delta_K}}{30}$)

Hard Problems in Class Goups

- Computation of $h(\mathcal{O}_{\Delta_K})$, the structure of $C(\mathcal{O}_{\Delta_K})$ and DL
- Sub exponential algorithm Hafner and McCurley, JoAMS'89
- Heuristic Complexity $L_{|\Delta_K|}[1/2, 1 + o(1)]$
- Smaller parameters than factoring

1827 bits Δ_K vs 3072 bits N

Vintage Crypto in $C(\mathcal{O}_{\Delta_K})$

- Diffie-Hellman key exchange and ElGamal Buchmann and Williams, JoC'88
- DSA and GQ signatures adaptations : Biehl, Buchmann, Hamdy, Meyer (01-02)
- Public coin setup:
  - $\Delta_k := -q$, $q \equiv 3 \pmod 4$, q prime : $h(\mathcal{O}_{\Delta_K})$ is odd
  - Choose g a random class of $C(\mathcal{O}_{\Delta_K})$
    $\rightsquigarrow$ order of g will be close to $h(\mathcal{O}_{\Delta_K}) \approx \sqrt{|\Delta_K|}$
  - Work in the cyclic group $G = \langle g \rangle \subset C(\mathcal{O}_{\Delta_K})$
- The order of g is unknown!

Paradox of Unknown Order 😕

- DL in a cyclic group $G = \langle g \rangle \subset C(\mathcal{O}_{\Delta_K})$ of unknown order s
- s is divisible by small primes with non negligible probability!
- But s not smooth for cryptographic sizes
- Many technicalities

Paradox of Unknown Order 😎

Public coin setup to generate a group of unknown order!

- Cryptographic accumulators Lipmaa, ACNS'12
- Verifiable delay functions Wesolowski, EC'19
- Transparent SNARKS Bünz, Fisch, Szepieniec, EC'20
- Range Proofs Couteau, Klooß, Lin, Reichle, EC'21
- ...

- RSA based construction: someone knows $\varphi(N)$! Needs some trusted setup.
- With class groups, $h(\mathcal{O}_{\Delta_K})$ is really unknown to anyone!

- Another application: LHE modulo a prime $\rightsquigarrow$ the CL cryptosystem

Outline

A Subgroup with an Easy DL

- C. Laguillaumie CT-RSA'15
- $\Delta_K = -pq$, $\Delta_q = -pq^3$, p, q primes and $pq \equiv 3 \pmod 4$

$$h(\mathcal{O}_{\Delta_q}) = q \times h(\mathcal{O}_{\Delta_K})$$

- Let $f := \left[q^2\mathbf{Z} + \frac{-q+\sqrt{\Delta_q}}{2}\mathbf{Z}\right] \in \mathrm{C}(\mathcal{O}_{\Delta_q})$
- $\mathrm{F} = \langle f \rangle$ is of order q, and

$$f^m = \left[q^2\mathbf{Z} + \frac{-\mathrm{L}(m)q + \sqrt{\Delta_q}}{2}\mathbf{Z}\right] \quad \text{where} \quad \mathrm{L}(m) \equiv 1/m \pmod q$$

CL Framework

Group with an easy DL subgroup

- q a prime
- $\mathrm{G} = \langle g \rangle$ cyclic group of order $q \cdot s$ such that $\gcd(q, s) = 1$
- $\mathrm{F} = \langle f \rangle$ subgroup of G of order q
- $\mathrm{G}^q = \langle h \rangle = \{x^q, x \in \mathrm{G}\}$ subgroup of G of order s,

$$\mathrm{G} \simeq \mathrm{F} \times \mathrm{G}^q$$

- DL is easy in F:

Given $u \in \mathrm{F}$, find $m \in \mathbf{Z}/q\mathbf{Z}$ such that $u = f^m$

CL Framework

Group with an easy DL subgroup

- q a prime
- $\mathrm{G} = \langle g \rangle$ cyclic group of order $q \cdot s$ such that $\gcd(q, s) = 1$
- $\mathrm{F} = \langle f \rangle$ subgroup of G of order q
- $\mathrm{G}^q = \langle h \rangle = \{x^q, x \in \mathrm{G}\}$ subgroup of G of order s,

$$\mathrm{G} \simeq \mathrm{F} \times \mathrm{G}^q$$

- Hard to distinguish elements of G^q:

$$\{\mathrm{Z} \hookleftarrow \mathrm{G}\} \approx_c \{\mathrm{Z} \hookleftarrow \mathrm{G}^q\}$$

Hard Subgroup Membership Assumption (HSM)

CL Framework

Group with an easy DL subgroup

- q a prime
- $G = \langle g \rangle$ cyclic group of order $q \cdot s$ such that $\gcd(q, s) = 1$
- $F = \langle f \rangle$ subgroup of G of order q
- $G^q = \langle h \rangle = \{x^q, x \in G\}$ subgroup of G of order s,

$$G \simeq F \times G^q$$

- Inspired by Camenisch, Shoup C'03 / Bresson , Catalano, Pointcheval AC'03 (03) : constructions over Paillier $\langle 1 + N \rangle \subset (\mathbf{Z}/N^2\mathbf{Z})^\times$

CL(-HSM) LHE

- $\mathscr{M} = \mathbf{Z}/q\mathbf{Z}$
- KeyGen:

  $sk = x \xleftarrow{\$} \{1, \dots, B\}$
  $pk \leftarrow h^x$
- Encrypt:

  $r \xleftarrow{\$} \{1, \dots, B\}$
  $c = (c_1, c_2) \leftarrow (h^r, f^m\, pk^r)$
- Decrypt:

  $\mathsf{DL}_f(c_2/c_1^x) \rightsquigarrow m$

- $c \otimes c' := (c_1 c_1', c_2 c_2') = (h^{r+r'}, f^{m+m'}\, pk^{r+r'})$
- $c^{\otimes \alpha} := (c_1^\alpha, c_2^\alpha) = (h^{r\alpha}, f^{m\alpha}\, pk^{r\alpha})$

Outline

Distributed Key Generation for Elgamal

Elgamal Keygen: $sk = x \xleftarrow{\$} \mathbf{Z}/q\mathbf{Z} \quad pk \leftarrow h^x$

DKG for Elgamal Pedersen EC'91

- Party i:
  - samples α_i
  - publishes h^{α_i}
  - acts as a Dealer for a verifiable additive secret sharing: $\alpha_i \rightsquigarrow (\alpha_{i,1}, \dots \alpha_{i,n})$
- Party i sums the shares received from the qualified parties: $\gamma_i := \sum_{j \in \mathscr{Q}} \alpha_{j,i}$

  $\rightsquigarrow$ secret sharing of implicit $sk := \sum_{i \in \mathscr{Q}} \alpha_i$
- Set $pk := \prod_{i \in \mathscr{Q}} h^{\alpha_i}$
- Computations modulo the known order of h

Moving to CL

CL Keygen: $sk = x \xleftarrow{\$} \{1, \dots, B\}$ $\quad pk \leftarrow h^x$

DKG for CL

Braun, Damgård, Orlandi, C'23 ; Braun, C., Damgård, Laguillaumie, Melissaris, Orlandi, Tucker, SCN'24

- Unknown order: computations over $\mathbf{Z}$
- Use Shamir's secret sharing over $\mathbf{Z}$: $\alpha_{i,j} := f_i(j)$
- Feldman's verification of the received shares (evaluation of f_i in the exponent)
- Weak Feldman reconstruction: Sharing α only assures to reconstruct $\bar{\alpha} \in \mathbf{Z}$ s.t. $h^{\bar{\alpha}} = h^{(n!)^2 \cdot \alpha}$

$$\rightsquigarrow \text{implicit } sk = (n!)^2 \cdot \sum_{i \in \mathcal{Q}} \alpha_i$$

- Pedersen DKG allows the adversary to bias the key: CL still IND-CPA
- Efficient ZKP using the Rough Order Assumption

Distributed Decryption for CL

CL Decryption: $\mathrm{DL}_f(c_2/c_1^{sk}) \rightsquigarrow m$

Distributed computation of c_1^x

- Qualified Party i computes and publishes $c_1^{(n!)^2 \cdot \gamma_i}$ (+ZKP)
- Combination of the results using Lagrange coefficient allows to get $\mathrm{W} = c_1^{(n!)^2 \cdot sk}$
- Compute

$$\mathrm{DL}_f(c_2^{(n!)^2}/\mathrm{W}) \rightsquigarrow (n!)^2 \cdot m$$

- Defined modulo the known prime q

We can recover m !

BICYCL

- BICYCL Implements CryptographY in CLass groups
- C++ library C., Bouvier, Imbert, Laguillaumie, JoC'23
- Efficient implementation of exponentiations:
  - NUCOMP, NUDUPL Shanks NATO NTA'89
  - Inversion are free $\rightsquigarrow$ adapt exponentiation optimizations of ECC
  - Fixed base exponentiation

CL versus Paillier

Sec. level		CL	Paillier
112	ciphertext	2694 bits	4096 bits
	setup	0.300 s	-
	keygen	0.011 s	0.039 s
	encrypt	4.39 ms	6.57 ms
	decrypt	9.70 ms	6.56 ms
128	ciphertext	3509 bits	6144 bits
	setup	0.586 s	-
	keygen	0.019 s	0.121 s
	encrypt	7.68 ms	20.9 ms
	decrypt	17.8 ms	20.8 ms

Threshold CL

	n	10	100	1000
DKG	Dealing	24	241	17 325
	Check	63	3 609	351 540
	Extract	7	467	46 338
	total comp.	93	4 318	415 203
	total comm.	2	30	1 849
Decryption	Partial Dec.	13	32	352
	Verify	14	28	253
	Combine	2	59	6 922
	total comp.	29	119	7 527
	total comm.	1	1	2

- In ms/KB for a single party at 112 bits security with $t = n/2 - 1$

Threshold CL

- Ongoing “clean” implementation (better batched ZKP) by Bouvier and Combal
- Stay tuned on

`https://gite.lirmm.fr/crypto/bicycl`