Updatable Privacy-Preserving Blueprints

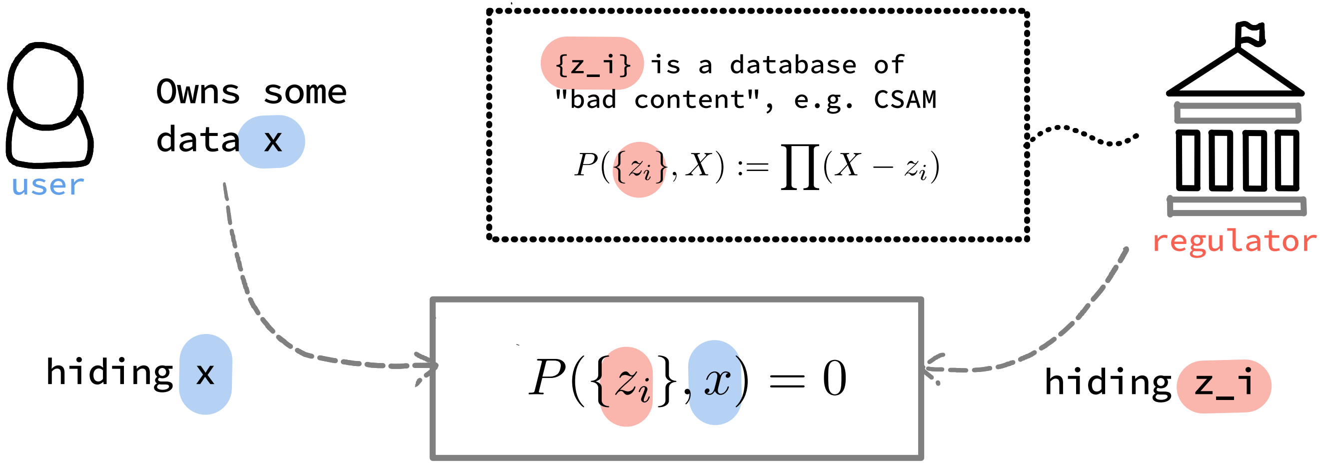
Bernardo David, Felix Engelmann, Tore Frederiksen,
Markulf Kohlweiss, Elena Pagnin, and Mikhail Volkhov

O1Labs

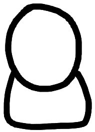
misha@o1labs.org

ia.cr/2023/1787

Regulation compliance & data scanning



Regulation compliance & data scanning

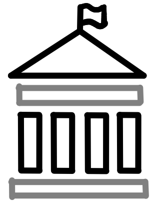


user

Owns some data x

{z_i} is a database of "bad content", e.g. CSAM

$$P(\{z_i\}, X) := \prod (X - z_i)$$



regulator

hiding x

$$P(\{z_i\}, x) = 0$$

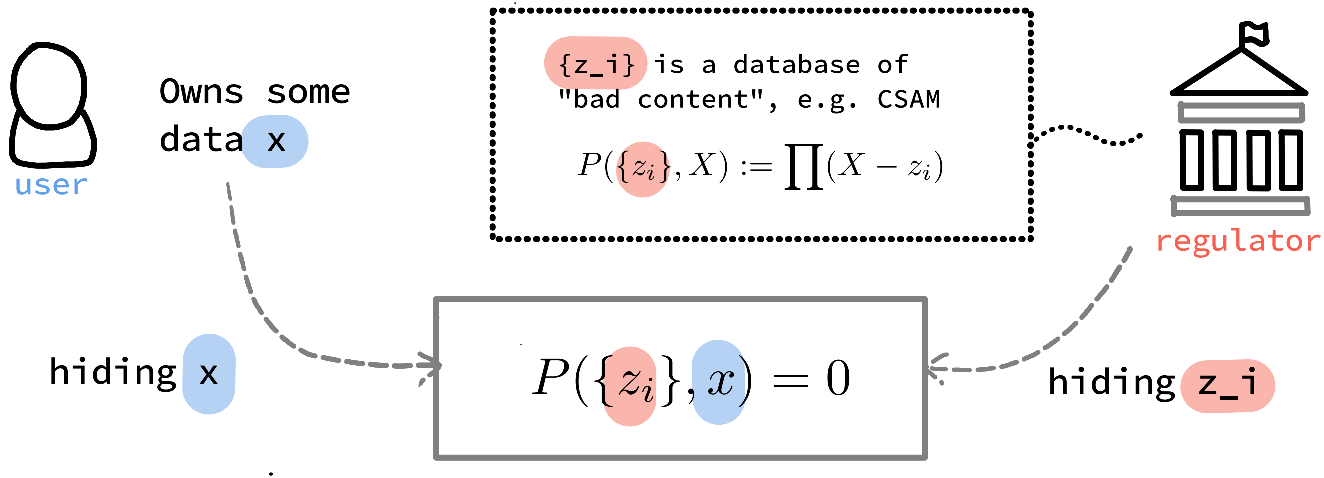
hiding z_i

Solutions:

1. U sends the data to X, or the provider must store it

FATF approved but not private

Regulation compliance & data scanning

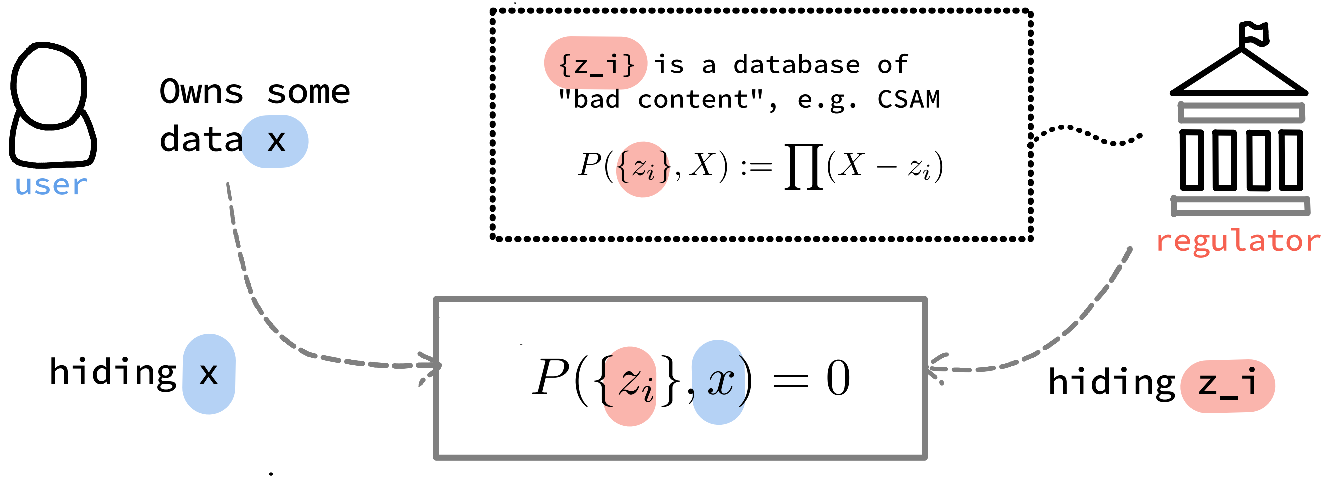


Solutions:

1. U sends the data to X, or the provider must store it — FATF approved but not private

2. General interaction-heavy MPC — e.g. SPDZ [DPSZ12] "from somewhat homomorphic enc"

Regulation compliance & data scanning



Solutions:

1. U sends the data to X, or the provider must store it   FATF approved but not private

2. General interaction-heavy MPC e.g. SPDZ [DPSZ12] "from somewhat homomorphic enc"

3. Client-side scanning: requires "authority code" running on the client, native or in SGX. Prone to leaking z.

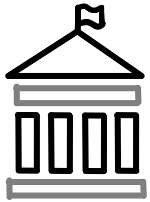
"Bugs in our Pockets" by Abelson et al.

Privacy-Preserving Blueprints [KLN22]

Eurocrypt23
Kohlweiss, Lysyanskaya, An Nguyen

$$P(\{z_i\}, X) := \prod (X - z_i)$$

$\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}}$

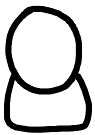


regulator

encrypted "blueprint" containing $\{z_i\}$

{zi} are committed & cannot be changed on the fly

x



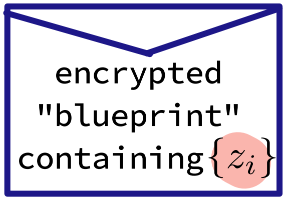
user

knows $\mathsf{pk}_{\mathcal{R}}$

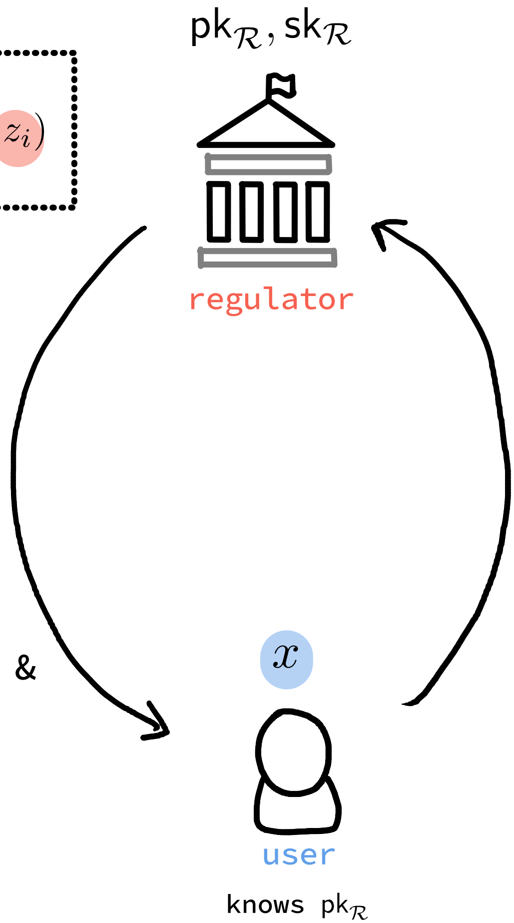
Privacy-Preserving Blueprints [KLN22]

Eurocrypt23
Kohlweiss, Lysyanskaya,
An Nguyen

$$P(\{z_i\}, X) := \prod (X - z_i)$$



{zi} are committed & cannot be changed on the fly



escrow(x)

reveals only $[P(x, \{z_i\}) = 0]$

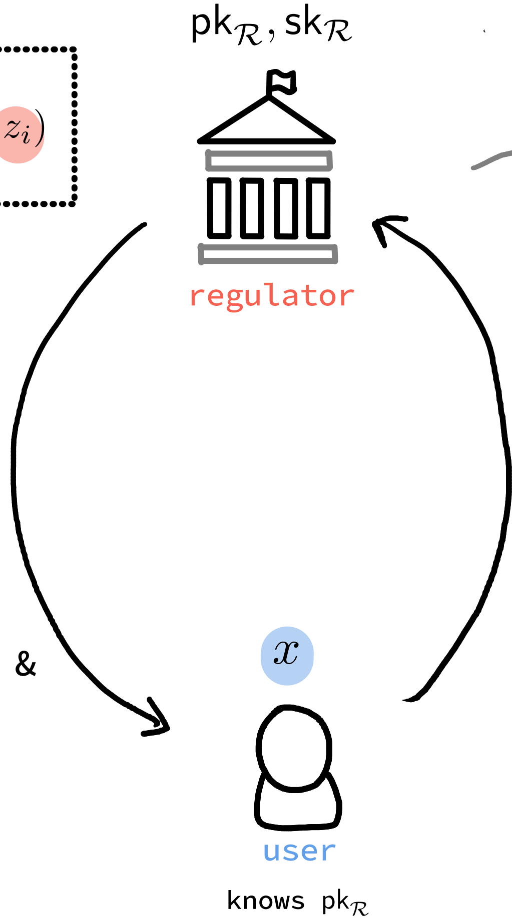
Privacy-Preserving Blueprints [KLN22]

Eurocrypt23
Kohlweiss, Lysyanskaya,
An Nguyen

$$P(\{z_i\}, X) := \prod (X - z_i)$$

encrypted
"blueprint"
containing $\{z_i\}$

{zi} are committed &
cannot be changed
on the fly



Verifies escrow
learns whether
$$P(\{z_i\}, x) = 0$$

escrow(x)

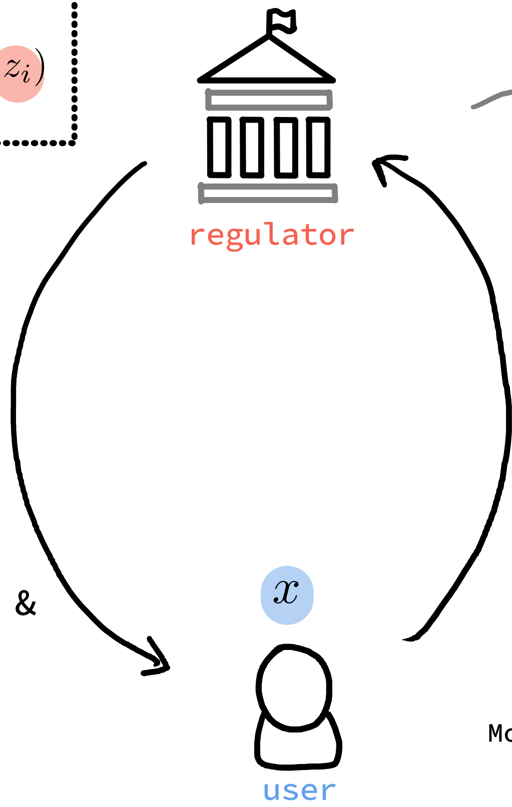
reveals only
$$[P(x, \{z_i\}) = 0]$$

Privacy-Preserving Blueprints [KLN22]

Eurocrypt23
Kohlweiss, Lysyanskaya, An Nguyen

$$P(\{z_i\}, X) := \prod (X - z_i)$$

$\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}}$



Verifies escrow

learns whether

$$P(\{z_i\}, x) = 0$$

encrypted "blueprint" containing $\{z_i\}$

escrow(x)

reveals only $[P(x, \{z_i\}) = 0]$

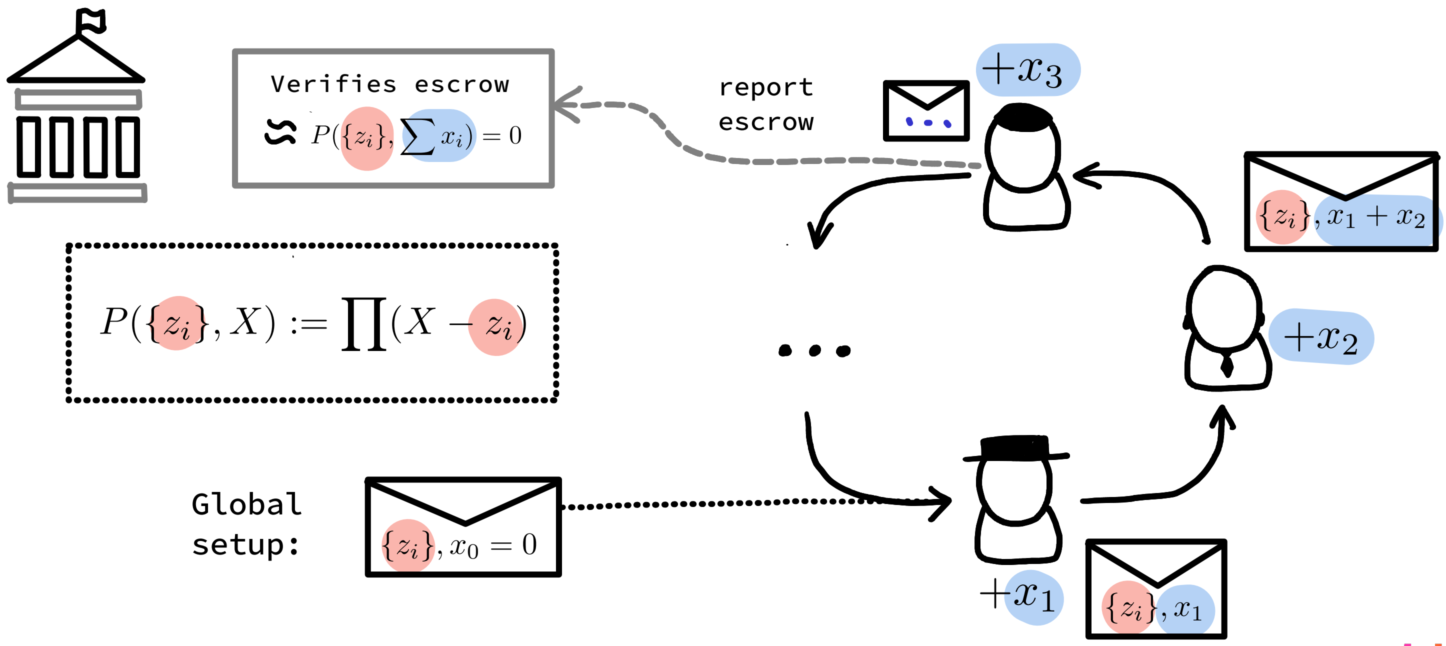
{zi} are committed & cannot be changed on the fly

More features in the next talk

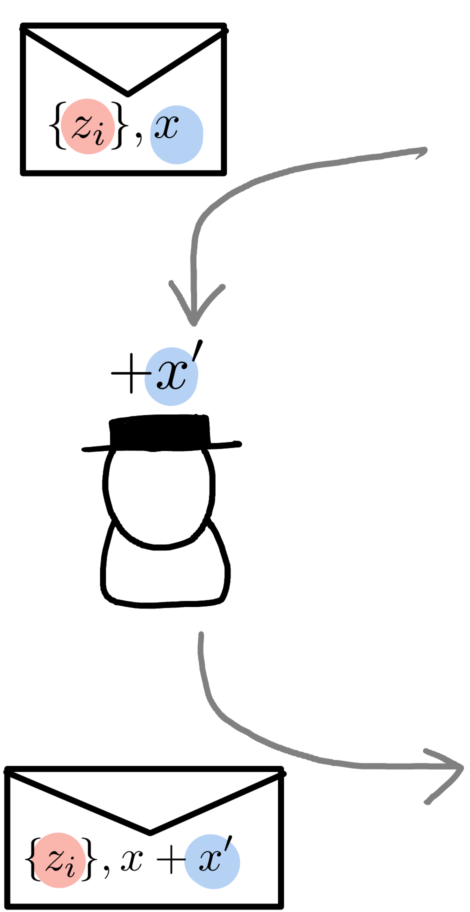
knows $\mathsf{pk}_{\mathcal{R}}$

Updatable Blueprints

Q: Can we allow users to update their secret?



Updatable Blueprints: Privacy Expectations



Things user knows:

- their own x'

Things user doesn't know:

- regulator's $\{z_i\}$
- previous step x
- whether

$$P(\{z_i\}, \sum x_i) = 0$$

More Applications

$$[t, \ldots, t + \Delta]$$

each $x_i \in [0, \Delta]$

$$P(X, t) = \prod_{i=0}^{\Delta} (X - (t + i))$$

More Applications

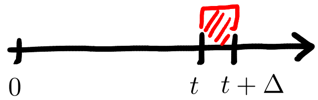
$$[t, \ldots, t + \Delta]$$

each $x_i \in [0, \Delta]$

$$P(X, t) = \prod_{i=0}^{\Delta} (X - (t + i))$$

1. Banks tracking rating for regulator
   (credit score/tx limits)

$$x + x_i \in [t, \ldots, t + \Delta]$$



More Applications

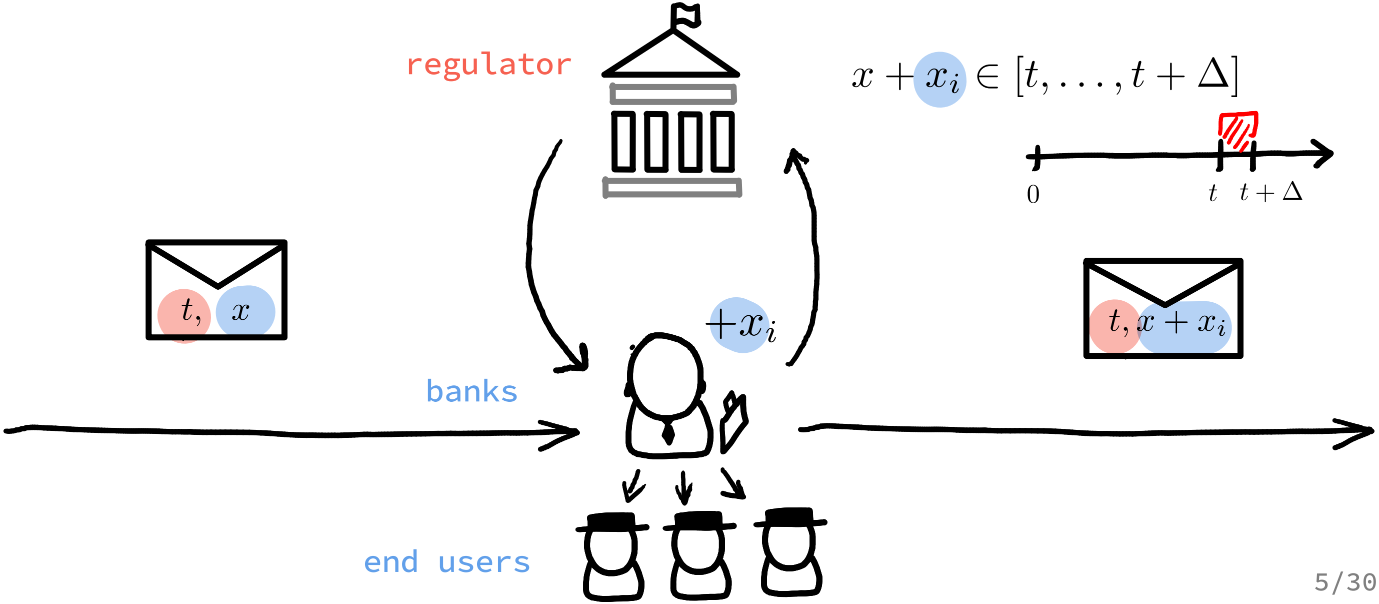
$$[t, \ldots, t + \Delta]$$

each $x_i \in [0, \Delta]$

$$P(X, t) = \prod_{i=0}^{\Delta} (X - (t + i))$$

1. Banks tracking rating for regulator (credit score/tx limits)

$$x + x_i \in [t, \ldots, t + \Delta]$$



More Applications

$$[t, \ldots, t+\Delta]$$

each $x_i \in [0, \delta]$

$$P(X, t) = \prod_{i=0}^{\Delta} (X - (t + i))$$

1. Banks tracking rating for regulator
   (credit score/tx limits)

2. Primitive voting

if $\delta = 1$
(one vote per party)

regulator learns $\underbrace{\sum^{n} x_i}_{\text{sum of votes}} > t$

More Applications

each $x, y \in [0, \delta]$

1. Banks tracking rating for regulator
   (credit score/tx limits)

2. Primitive voting

3. (Extension) Euclidian proximity testing
   e.g. covid prox. testing; each antenna communicates relative change only

regulator learns $$(\sum x - t_x)^2 + (\sum y - t_y)^2 > \Delta^2$$

Our Contributions

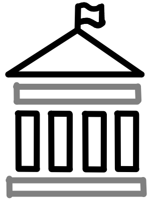
- Updatable blueprints
  - A novel primitive for MPC predicate checking
  - Regulator learns $P(t, \sum x_i,)$

    (user's x, regulator's t)

- Efficient instantiation for range predicates
  - Showcases updatable NIZKs

- Extendable to a large class of predicates

Syntax

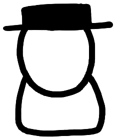


regulator

threshold

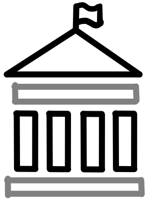
$$\mathsf{sk}, \mathsf{pk}, \mathsf{hint}_0 \leftarrow \mathsf{KeyGen}(P, t)$$

x_ι



user

Syntax



regulator

threshold

$$\mathsf{sk}, \mathsf{pk}, \mathsf{hint}_0 \leftarrow \mathsf{KeyGen}(P, t)$$

$\iota++$

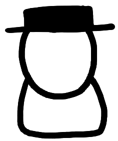
- hint is well-formed
- (at each prev step $x_\iota \in [0, \delta]$)

$$0/1 \leftarrow \mathsf{VfHint}_{\mathsf{pk}}(\mathsf{hint}_{\iota-1})$$

$$\mathsf{hint}_\iota \leftarrow \mathsf{Update}_{\mathsf{pk}}(\mathsf{hint}_{i-1}, x_\iota)$$

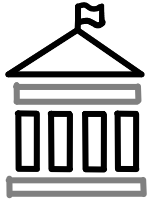
"contains" $\{t^i x^j\}$

x_ι



user

Syntax



regulator

threshold

$$\text{sk}, \text{pk}, \text{hint}_0 \leftarrow \text{KeyGen}(P, t)$$

$\iota++$

- hint is well-formed
- (at each prev step $x_\iota \in [0, \delta]$)

$$0/1 \leftarrow \text{VfHint}_{\text{pk}}(\text{hint}_{\iota-1})$$

$$\text{hint}_\iota \leftarrow \text{Update}_{\text{pk}}(\text{hint}_{i-1}, x_\iota)$$

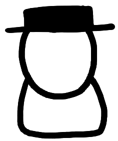
"contains" $\{t^i x^j\}$

esc_ι

$$\text{esc}_\iota \leftarrow \text{Escrow}_{\text{pk}}(\text{hint}_\iota)$$

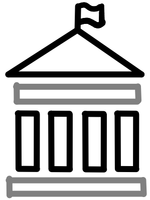
"contains" $[P(t, x) = 0]$

x_ι



user

Syntax



regulator

$0/1 \leftarrow \mathsf{VfEscrow}_{\mathsf{pk}}(\mathsf{esc}_\iota)$

$0/1 \leftarrow \mathsf{Decrypt}_{\mathsf{sk}}(\mathsf{esc}_\iota)$

decrypts to $[P(t, \sum x_i) = 1]$

threshold

$\mathsf{sk}, \mathsf{pk}, \mathsf{hint}_0 \leftarrow \mathsf{KeyGen}(P, t)$

esc_ι

$\iota++$

- hint is well-formed
- (at each prev step $x_\iota \in [0, \delta]$)

$0/1 \leftarrow \mathsf{VfHint}_{\mathsf{pk}}(\mathsf{hint}_{\iota-1})$

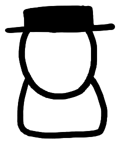
$\mathsf{esc}_\iota \leftarrow \mathsf{Escrow}_{\mathsf{pk}}(\mathsf{hint}_\iota)$

"contains" $[P(t, x) = 0]$

$\mathsf{hint}_\iota \leftarrow \mathsf{Update}_{\mathsf{pk}}(\mathsf{hint}_{i-1}, x_\iota)$

"contains" $\{t^i x^j\}$

x_ι



user

Construction Idea

(honest case)

1. Initial hints:

   ElGamal enc P in the exponent to regulator's pk

$$\mathsf{hint}_0 := \mathsf{Enc}_{\mathsf{pk}}(\text{"}P(t, 0)\text{"})$$

Construction Idea
(honest case)

1. Initial hints:
   ElGamal enc P in the exponent to regulator's pk

$$\mathsf{hint}_0 := \mathsf{Enc}_{\mathsf{pk}}(\text{"}P(t, 0)\text{"})$$

2. Each step updates hints

$$\mathsf{hint}_\iota \approx \mathsf{Enc}(P(t, x))$$
$$\downarrow \qquad \qquad \downarrow$$
$$\mathsf{hint}_{\iota+1} \approx \mathsf{Enc}(P(t, x + x_\iota))$$

Construction Idea
(honest case)

1. Initial hints:
   ElGamal enc P in the exponent to regulator's pk

$$\mathsf{hint}_0 := \mathsf{Enc}_{\mathsf{pk}}(\text{"}P(t, 0)\text{"})$$

2. Each step updates hints

$$\mathsf{hint}_\iota \approx \mathsf{Enc}(P(t, x))$$
$$\downarrow \qquad\qquad \downarrow$$
$$\mathsf{hint}_{\iota+1} \approx \mathsf{Enc}(P(t, x + x_\iota))$$

3. On query from R, randomise & report

$$\mathsf{Enc}(\beta \cdot P(t, x))$$

$$P(t, x) = 0 \implies \beta P(t, x) = 0$$
$$P(t, x) \neq 0 \implies \beta P(t, x) \approx \beta$$

Predicate polynomial

Threshold predicate for range [t, d+t]

$$P_d(T, X) = \left[\prod_{i=0}^{d-1} (X - (T + i)) \stackrel{?}{=} 0 \right] \in \{0, 1\}$$

Can be efficiently represented as:

$$\prod_{\delta=0}^{d-1} ((X - T) - \delta) = \sum_{i=0}^{d} U_i (X - T)^i$$

Where Ui are Stirling coefficients

Hints & P evaluation

Hints = ElGamal ciphertexts of powers in P(t,x):

$$\{A_i = G^{r_i}, B_i = G^{(x-t)^i} H^{r_i}\}_{i=0}^{d}$$

On KeyGen, the accumulated value x is 0

$$\{A_i = G^{r_i}, B_i = G^{(0-t)^i} H^{r_i}\}$$

Hints & P evaluation

Hints = ElGamal ciphertexts of powers in P(t,x):

$$\{A_i = G^{r_i}, B_i = G^{(x-t)^i} H^{r_i}\}_{i=0}^{d}$$

On KeyGen, the accumulated value x is 0

$$\{A_i = G^{r_i}, B_i = G^{(0-t)^i} H^{r_i}\}$$

Due to Stirling coeff representation we can eval P

$$\left(\prod A_i^{U_i}, \prod B_i^{U_i}\right) = \left(G^{\sum U_i r_i}, G^{\underbrace{\sum U_i (x-t)^i}} H^{\sum U_i r_i}\right)$$

$$= \mathsf{Enc}\left(P(t,x); r = \sum U_i r_i\right)$$

Escrow

reminder: hints

$$\{A_i = G^{r_i}, B_i = G^{(x-t)^i} H^{r_i}\}_{i=0}^{d}$$

Escrow is created by evaluating polynomial using hints with a randomizer

$$\left(\prod_{\delta=0}^{d} (A_i^{U_i})^{\beta}, \prod_{\delta=0}^{d} (B_i^{U_i})^{\beta}\right) = \mathsf{Enc}(\beta \cdot P(t, x))$$

Updating hints

- If Y is update value

$$(X - T + Y)^i = \sum_{j=0}^{i} \underbrace{\left(\binom{i}{j} Y^{i-j}\right)}_{\text{known to the User}} \cdot \underbrace{(X - T)^j}_{\text{stored in hints}}$$

olabs

Updating hints

- If Y is update value

$$(X - T + Y)^i = \sum_{j=0}^{i} \underbrace{\left(\binom{i}{j} Y^{i-j}\right)}_{\text{known to the User}} \cdot \underbrace{(X - T)^j}_{\text{stored in hints}}$$

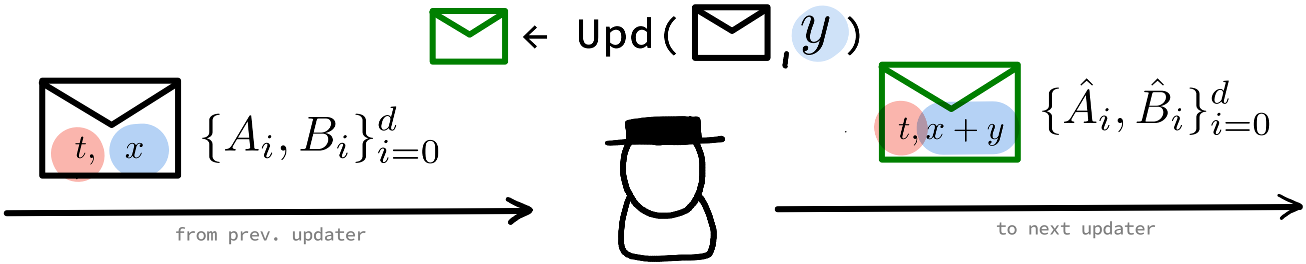
- This means hints can be updated too:

$$\{A_i, B_i\}_0^d \mapsto \{\hat{A}_i := A_i, \hat{B}_i := G^{y^i} \prod_{j=1}^{i} B_i^{\binom{i}{j} \cdot y^{i-j}}\}_0^d$$

accumulated value $(x - t)$ remains secret

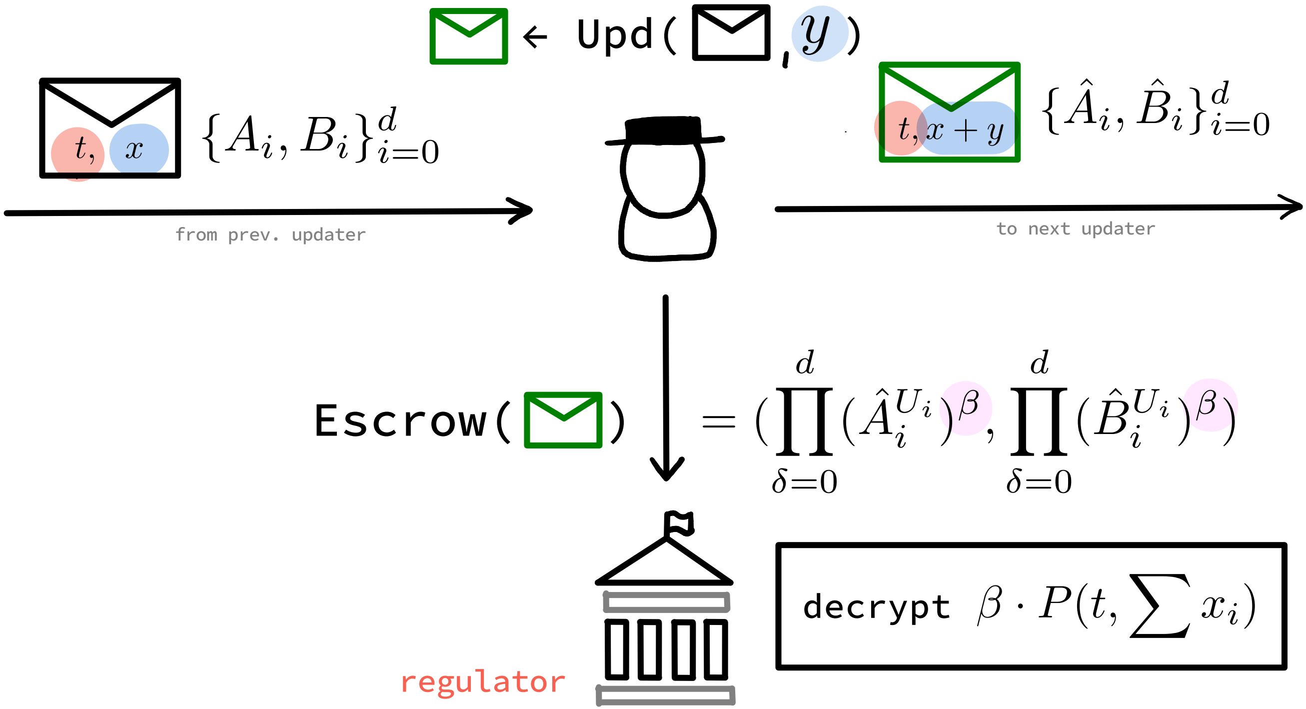
Honest construction with hints

$$\{\hat{A}_i := A_i, \hat{B}_i := G^{y^i} \prod_{j=1}^{i} B_i^{\binom{i}{j} \cdot y^{i-j}}\}$$



Honest construction with hints

$$\{\hat{A}_i := A_i, \hat{B}_i := G^{y^i} \prod_{j=1}^{i} B_i^{\binom{i}{j} \cdot y^{i-j}}\}$$



Achieving Privacy

against the regulator

Problem:

We will need to send hints to the Regulator for soundness, but hints reveal accumulated $x = \sum x_i$

up to last update

Achieving Privacy

against the regulator

Problem:

We will need to send hints to the Regulator for soundness, but hints reveal accumulated $x = \overset{\text{up to last update}}{\sum} x_i$

Solution: blind hints before escrowing

$$\{A_i, B_i\} \mapsto \{\hat{A}_i := A_i, \hat{B}_i := B_i \cdot W_i^{\alpha}\}$$

0 in hints

Achieving Soundness: NIZKs

1) Consistency of updates & escrow
   (proving hint was updated correctly)

2) Trace proof:
   Maintaining a history of updates
3) Key proof:
   Proving knowledge of t by regulator
4) Escrow proof:
   Building escrow encryption

Σ-protocols

Achieving Soundness: NIZKs

1) Consistency of updates & escrow
   (proving hint was updated correctly) } CH20

2) Trace proof:
   Maintaining a history of updates
3) Key proof:
   Proving knowledge of t by regulator
4) Escrow proof:
   Building escrow encryption

} Σ-protocols

Schnorr would be linear in #updates.
=> use updatable NIZKs

Controlled* Malleability in NIZKs

(updatability)

$$(x, w) \in \mathcal{R}$$

L.Update(T_x T_w)

$$(x', w') \in R$$

$$x' = T_x(x) \qquad w' = T_w(w)$$

* NB: Not to be confused with Controlled Malleability as a security notion

Controlled* Malleability in NIZKs

(updatability)

$$(x, w) \in \mathcal{R} \xrightarrow{\texttt{NIZK.Prove}} \pi \qquad x \in \mathcal{L}_{\mathcal{R}}$$

L.Update(T_x T_w)

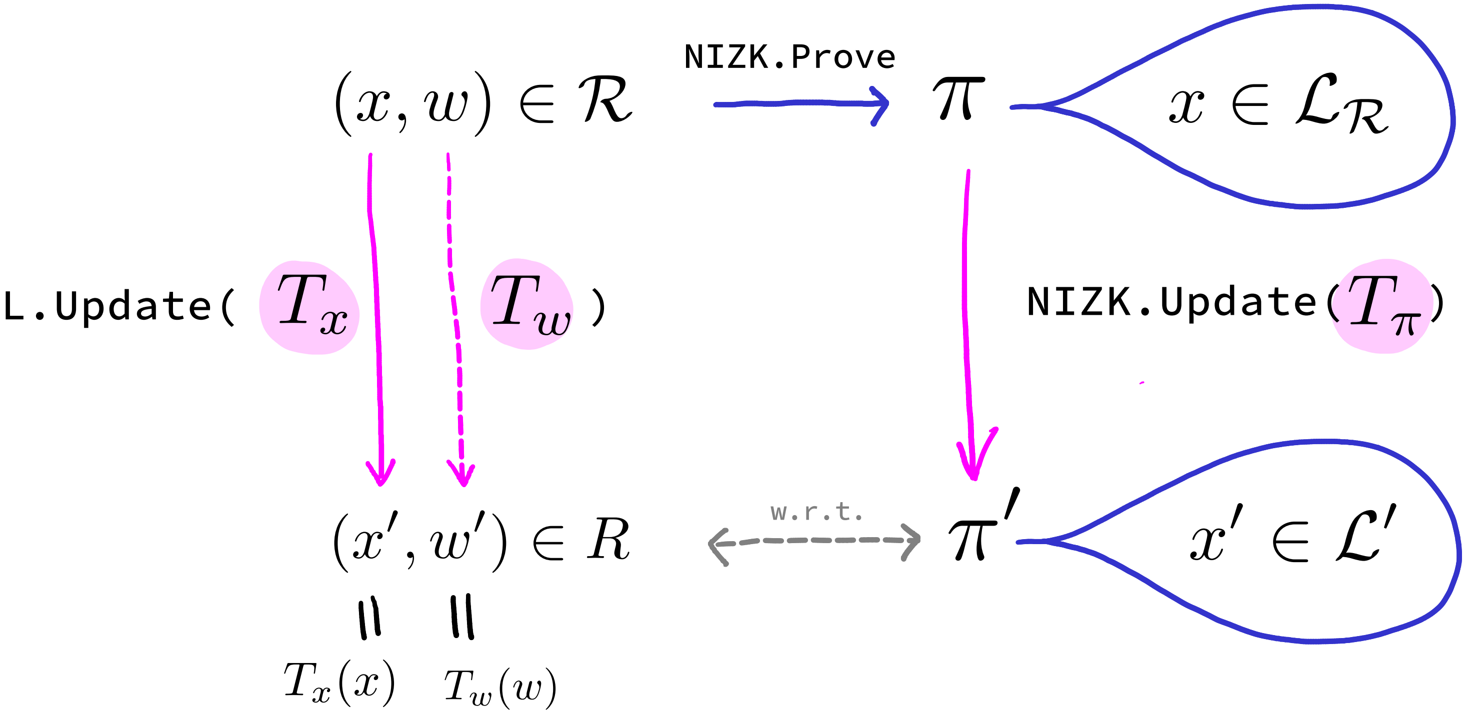
$$(x', w') \in R$$

$$x' = T_x(x) \qquad w' = T_w(w)$$

* NB: Not to be confused with Controlled Malleability as a security notion

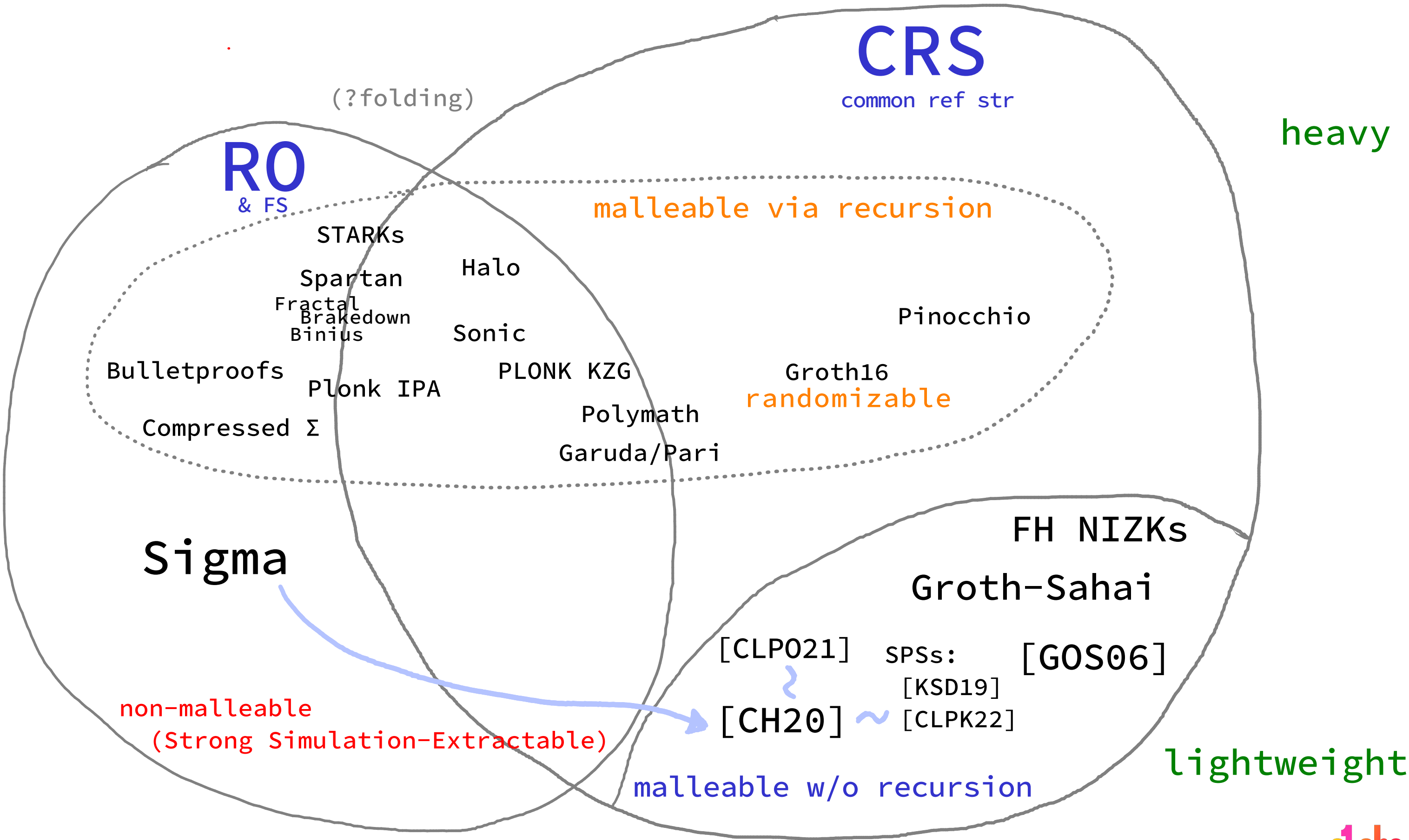
Controlled* Malleability in NIZKs

(updatability)



* NB: Not to be confused with Controlled Malleability as a security notion

Landscape of Malleable NIZKs



[CH20] is akin to the Σ-protocol

$$(\mathbf{M}, \mathbf{\Theta}) \xleftarrow{\$} \mathcal{D}_{par}$$

$$\begin{array}{lcl}
\mathcal{P} & & \mathcal{V} \\
[\mathbf{x}], \mathbf{w} & & [\mathbf{x}] \\
\mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^k & & \\
[\mathbf{a}] := [\mathbf{M}(\mathbf{x})]\mathbf{r} & \xrightarrow{\quad [\mathbf{a}] \quad} & \\
 & \xleftarrow{\quad e \quad} & e \xleftarrow{\$} \mathbb{Z}_p \\
\mathbf{d} := e\mathbf{w} + \mathbf{r} & \xrightarrow{\quad \mathbf{d} \quad} & \\
 & & \text{check} \\
 & & [\mathbf{M}(\mathbf{x})]\mathbf{d} \stackrel{?}{=} [\mathbf{\Theta}(\mathbf{x})]e + [\mathbf{a}]
\end{array}$$

For the algebraic language:

$$\mathcal{L}_{\mathsf{alg}} = \{\vec{x} \in \mathbb{G}^l \mid \exists \vec{w} \in \mathbb{Z}_p^t : M(\vec{x}) \cdot \vec{w} = \vec{x}\}$$

where $M(\vec{X}) \in \mathcal{P}^{l \times t}$

Couteau, Geoffroy, and Dominik Hartmann.
"Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages."
CRYPTO 2020.

CH20 NIZK

... but done with pairings

$\underline{\text{CRSGen}\ (1^\lambda):}$

$par := \mathcal{PG} \xleftarrow{\$} PGGen(1^\lambda)$

$e \xleftarrow{\$} \mathbb{Z}_p$

$\text{CRS} := (\mathcal{PG}, [e]_2), \mathcal{T} := e$

$\text{return } (par, \text{CRS}, \mathcal{T})$

$\underline{\text{Prove}\ (\text{CRS}, ([\mathbf{M}]_1, [\mathbf{\Theta}]_1), [\mathbf{x}]_1 \in \mathbb{G}_1^l, \mathbf{w} \in \mathbb{Z}_p^t):}$

$\mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^t$

$[\mathbf{a}]_1 := [\mathbf{M}(\mathbf{x})]_1 \mathbf{r}$ $\quad \in \mathbb{G}_2^t$

$[\mathbf{d}]_2 := [e]_2 \mathbf{w} + [\mathbf{r}]_2$ $\quad \in \mathbb{G}_1^l$

$\text{return } \sigma := \underbrace{([\mathbf{a}]_1, [\mathbf{d}]_2)}_{\pi}$

$\underline{\text{Verify}\ (\text{CRS}, ([\mathbf{M}]_1, [\mathbf{\Theta}]_1), [\mathbf{x}]_1, \sigma = ([\mathbf{a}]_1, [\mathbf{d}]_2)):}$

check

$$[\mathbf{M}(\mathbf{x})]_1 \bullet [\mathbf{d}]_2 \stackrel{?}{=} [\mathbf{\Theta}(\mathbf{x})]_1 \bullet [e]_2 + [\mathbf{a}]_1 \bullet [1]_2$$

Couteau, Geoffroy, and Dominik Hartmann.
"Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages."
CRYPTO 2020.

CH20 NIZK is updatable!

observed in [CLPK22]* for a variant of CH20

π

Define $\mathsf{Update}(([\boldsymbol{a}]_1, [\boldsymbol{d}]_2), T := (T_{\mathsf{am}}, T_{\mathsf{aa}}, T_{\mathsf{xm}}, T_{\mathsf{xa}}, T_{\mathsf{wm}}, T_{\mathsf{wa}}))$ as a function returning $\pi' = ([\boldsymbol{a}']_1, [\boldsymbol{d}']_2)$ constructed as follows:

$\hat{\pi}$

$$[\boldsymbol{a}']_1 = T_{\mathsf{am}} \cdot \begin{pmatrix} [\boldsymbol{a}]_1 \\ \mathsf{x} \end{pmatrix} + [1]_1 \cdot T_{\mathsf{aa}} + [M(\mathsf{x}')]_1 \cdot \hat{\boldsymbol{s}}$$

$$[\boldsymbol{d}']_2 = T_{\mathsf{wm}} \cdot [\boldsymbol{d}]_2 + [z]_2 \cdot T_{\mathsf{wa}} + [1]_2 \cdot T_{\mathsf{wa}} + [1]_2 \cdot \hat{\boldsymbol{s}}$$

where $\hat{\boldsymbol{s}}$ is sampled uniformly at random.

...for blinding-compatible transformations

new notion necessary to achieve
proof updatability

* Improved Constructions of Anonymous Credentials From Structure-Preserving Signatures on Equivalence Classes
Aisling Connolly1, Pascal Lafourcade, and Octavio Perez Kempner

Transformations and Blinding-Compatibility

Let $(T_{\mathsf{xm}}, T_{\mathsf{xa}}, T_{\mathsf{wm}}, T_{\mathsf{wa}})$ be a valid language transformation:

$$\underset{x=[M]\cdot w}{(x,w)\in\mathcal{R}} \implies (T_{\mathsf{xm}}\cdot x + T_{\mathsf{xa}}, T_{\mathsf{wm}}\cdot w + T_{\mathsf{wa}}) \in \mathcal{R}$$

Could we transform the proof like this?

looks like instance? $[\vec{a}]_1 = [M]_1\vec{r}$

looks like witness? $[\vec{d}]_2 = [e]_2 \cdot \vec{w} + [\vec{r}]_2$

$$\implies$$

$$[\vec{a}']_1 = T_{\mathsf{xm}}\cdot[\vec{a}]_1 + T_{\mathsf{xa}}$$
$$[\vec{d}']_2 = T_{\mathsf{wm}}\cdot[\vec{d}]_2 + T_{\mathsf{wa}}$$

Transformations and Blinding-Compatibility

Let $(T_{\mathsf{xm}}, T_{\mathsf{xa}}, T_{\mathsf{wm}}, T_{\mathsf{wa}})$ be a valid language transformation:

$$\underbrace{(x, w) \in \mathcal{R}}_{x = [M] \cdot w} \implies (T_{\mathsf{xm}} \cdot x + T_{\mathsf{xa}}, T_{\mathsf{wm}} \cdot w + T_{\mathsf{wa}}) \in \mathcal{R}$$

Could we transform the proof like this?

looks like instance? $[\vec{a}]_1 = [M]_1 \vec{r}$

looks like witness? $[\vec{d}]_2 = [e]_2 \cdot \vec{w} + [\vec{r}]_2$

$$\implies \quad \begin{aligned} [\vec{a}']_1 &= T_{\mathsf{xm}} \cdot [\vec{a}]_1 + T_{\mathsf{xa}} \\ [\vec{d}']_2 &= T_{\mathsf{wm}} \cdot [\vec{d}]_2 + T_{\mathsf{wa}} \end{aligned}$$

This does not work! $([a]_1, [d]_2)$ are unlike a proper inst/wit because witness is not uniformly distributed!

Therefore we require T to be blinding-compatible: $\exists T_{\mathsf{am}}, T_{\mathsf{aa}}. \forall x \in \mathcal{L}, \forall s$

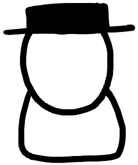
uniform; pseudo-witness

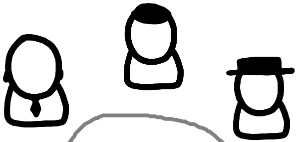
$$T_{\mathsf{am}} \cdot \begin{pmatrix} M(\vec{x}) \cdot \vec{s} \\ \vec{x} \end{pmatrix} + T_{\mathsf{aa}} = M(T_{\mathsf{xm}} \cdot \vec{x}) + T_{\mathsf{xa}} \cdot \left(T_{\mathsf{wm}} \cdot \vec{s} + T_{\mathsf{wa}}\right)$$

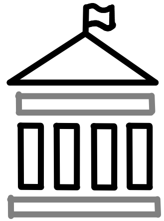
instance

pseudo-instance

UBlu with updatable NIZKs







regulator learns:

whether $P(t, \hat{x}) = 0$

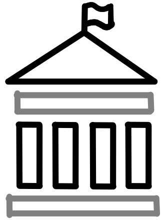
ElGamal

$$\{\mathsf{Enc}_{\mathsf{pk}}(x^i t^j)\} \xRightarrow{\text{update}} \{\mathsf{Enc}_{\mathsf{pk}}(\hat{x}^i t^j)\} \xRightarrow{\text{eval}} \mathsf{Enc}_{\mathsf{pk}}(\beta \cdot P(t, \hat{x}))$$

where

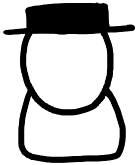
$$\hat{x} = x + x_\iota$$

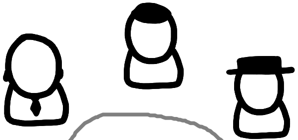
UBlu with updatable NIZKs



regulator learns:

whether $P(t,\hat{x}) = 0$





update: $\pi \mapsto \hat{\pi}$

ElGamal

$$\{\mathsf{Enc}_{\mathsf{pk}}(x^i t^j)\} \overset{\text{update}}{\Longrightarrow} \{\mathsf{Enc}_{\mathsf{pk}}(\hat{x}^i t^j)\} \overset{\text{eval}}{\Longrightarrow} \mathsf{Enc}_{\mathsf{pk}}(\beta \cdot P(t,\hat{x}))$$

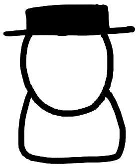
π $\qquad$ $\hat{\pi}$ $\qquad$ $\hat{\pi}$ verifies

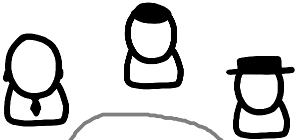
where

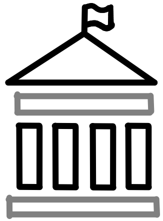
$$\hat{x} = x + x_\iota$$

Use CH20 to prove consistency of update/eval

UBlu with updatable NIZKs







regulator learns:

whether $P(t, \hat{x}) = 0$

$\pi \mapsto \hat{\pi}$

ElGamal   update   eval

$$\{\mathsf{Enc}_{\mathsf{pk}}(x^i t^j)\} \Longrightarrow \{\mathsf{Enc}_{\mathsf{pk}}(\hat{x}^i t^j)\} \Longrightarrow \mathsf{Enc}_{\mathsf{pk}}(\beta \cdot P(t, \hat{x}))$$

π $\hat{\pi}$ $\hat{\pi}$ verifies

where

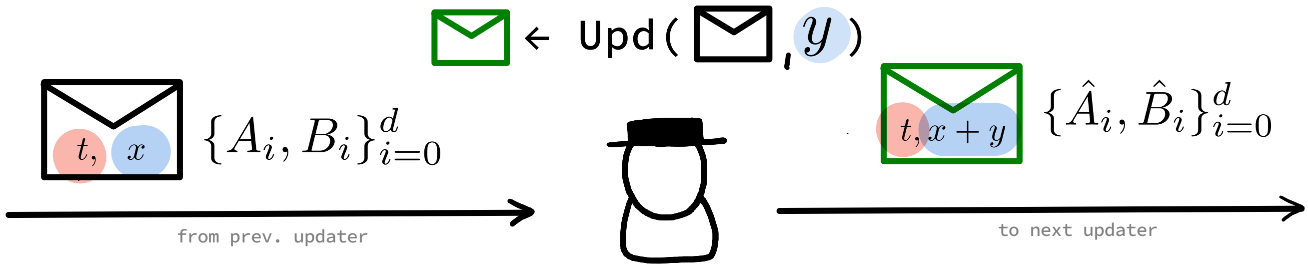
$$\hat{x} = x + x_\iota$$

Use CH20 to prove consistency of update/eval

$$\mathfrak{T} = G^{t} \mathfrak{H}^{r_{\mathsf{t}}} \qquad \mathfrak{X} = G^{\hat{x}} \mathfrak{H}^{\hat{r}_{\mathsf{x}}}$$

$$\mathfrak{A} = G^{\alpha} \mathfrak{H}^{r_{\alpha}} \qquad (A_i, D_i) = (G^{\hat{r}_i}, G^{(\hat{x}-t)^i} H^{\hat{r}_i} W_i^{\alpha}) \text{ for } i \in [d]$$

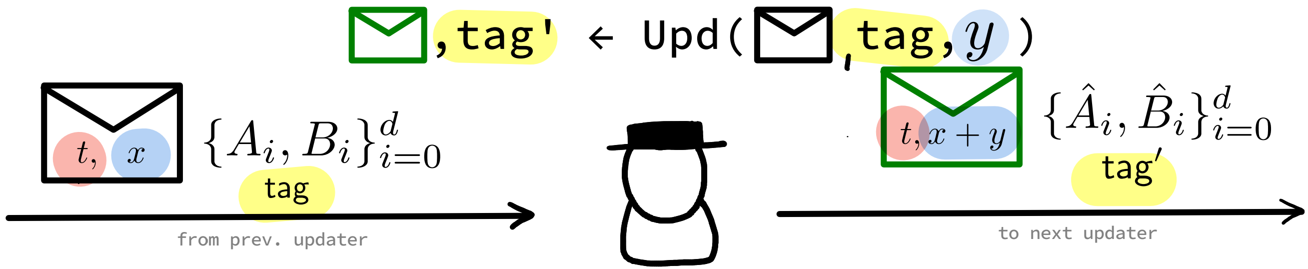
History: Tracking updates



Problem:

Right now we cannot reason about
"update number i" and order in general
since hints at each step look exactly the same

History: Tracking updates



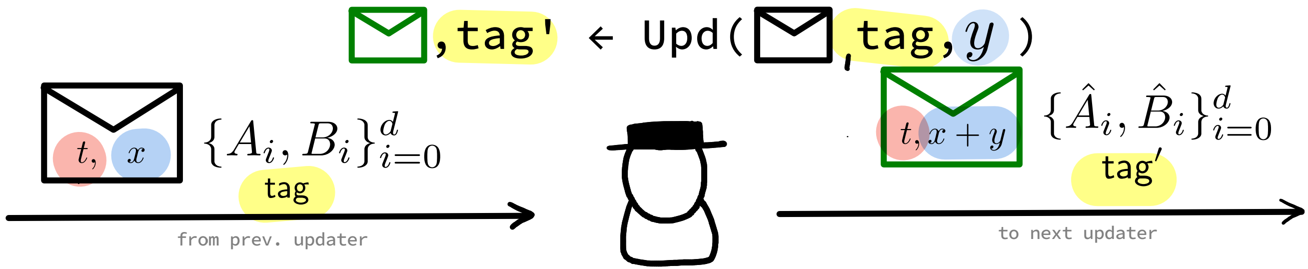
$\{\text{tag}_i\}_0^\iota$ is a chain of lightweight "update receipts"

$$\text{tag}_\iota = (\pi_\iota, X_\iota := \text{Com}(\sum^{\iota} x_i))$$

Schnorr proof of

$$X_\iota = X_{\iota-1} \cdot G^{x_\iota} H^r$$

History: Tracking updates



$\{\mathsf{tag}_i\}_0^\iota$ is a chain of lightweight "update receipts"

$$\underbrace{\mathsf{tag}_\iota = (\pi_\iota, X_\iota}_{\text{Schnorr proof of}} := \mathsf{Com}(\sum^{\iota} x_i))$$

$$X_\iota = X_{\iota-1} \cdot G^{x_\iota} H^r$$

Then, $\mathsf{VfHistory}(\{\mathsf{tag}_\iota\}) \rightarrow \{0,1\}$ checks all proofs

tags are hiding and can be put on the bulletin board

Security Properties

➤ Soundness

- Verify History ➡ extract update values $\{x_i\}$
  from straightline-extractable history proofs
- Verify Hint & Escrow ➡ $\mathrm{Dec} = [P(t, \sum x_i) \stackrel{?}{=} 0]$
  with history tags

note: [CH20] is only Sound, we can't extract

➤ History binding

Updates produce tags that "bind" $\{x_i\}$

- History verifies ➡ prefix verifies
- One can't produce alternative verifying history that has different tags in the middle, but same suffix&prefix

➤ Hiding x4: Threshold, history, tags, escrow

All: game-based definitions, under DDH & NIZK assumptions
(variant of kerMDH, falsifiable)

Performance: Size

Object	pp	sk	pk	hint	tag	esc
$\#\mathbb{G}_1$	$O(d^2)$	1	$O(1)$	$4d + 5$	$O(1)$	$4d + O(1)$
$\#\mathbb{G}_2$	$O(1)$	0	0	$2d + 8$	0	$2d + 8$

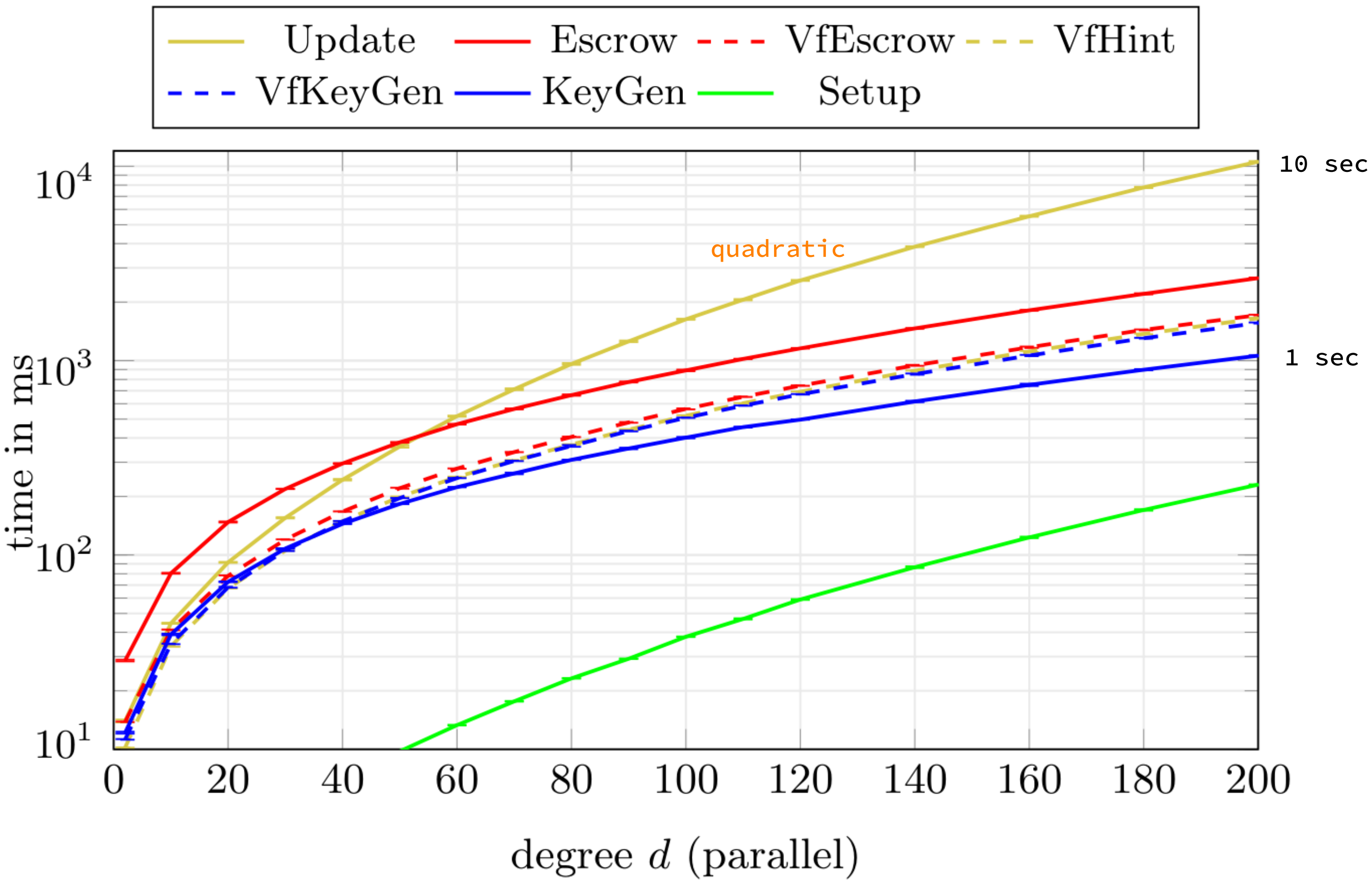
linear only in the degree
of the predicate polynomial
$P(T, X)$

Performance: Asymptotic time

Algorithm	$\#P$	$\#E_1$	$\#E_2$
Setup	0	$O(1)$	$O(1)$
KeyGen	0	$9d + O(1)$	$4d + O(1)$
Update	0	$4d^2 + O(d)$	$1.5d^2 + O(d)$
Escrow	0	$14d + O(1)$	$4d + O(1)$
Decrypt	0	$O(1)$	0
VfKeyGen	$2d + O(1)$	$10d + O(1)$	$6d + O(1)$
VfHint	$2d + O(1)$	$10d + O(1)$	$6d + O(1)$
VfHistory	0	$O(\iota_{\mathsf{cur}})$	$O(\iota_{\mathsf{cur}})$
VfEscrow	$2d + O(1)$	$10d + O(1)$	$6d + O(1)$

CH20
proof update

$\{(x-t)^i\}_0^d \mapsto \{(x+y-t)^i\}_0^d$ is inherently quadratic



Xeon E-2286G CPU @ 4 GHz; 6 cores, 12 threads

Extensions

- Arbitrary polynomial predicates:

  Note: hints are consistent powers $(x-t)^i$

  With quadratic number of hints: $x^i t^j$ with updates

$$(x+y)^i t^j = \sum_{k=0}^{i} \binom{i}{k} y^{i-k} (t^j x^k)$$

  Escrow polynomial with given coefficients

Extensions

- Arbitrary polynomial predicates:

  Note: hints are consistent powers $(x-t)^i$

  With quadratic number of hints: $x^i t^j$ with updates

  $$(x+y)^i t^j = \sum_{k=0}^{i} \binom{i}{k} y^{i-k} (t^j x^k)$$

  Escrow polynomial with given coefficients

- Multivariate polynomials: Run construction in parallel & bind with commitments

Extensions

- Arbitrary polynomial predicates:

  Note: hints are consistent powers $(x-t)^i$

  With quadratic number of hints: $x^i t^j$ with updates

  $$(x+y)^i t^j = \sum_{k=0}^{i} \binom{i}{k} y^{i-k} (t^j x^k)$$

  Escrow polynomial with given coefficients

- Multivariate polynomials:

  Run construction in parallel
  & bind with commitments

- Non-binary outputs:

  Binary predicate is $\beta P(t,x)$

  Secondary value on success:

  returns either (rand, rand)
  or (0, P_2(t,x))

  $$(\beta_1 P(t,x), \beta_2 P(t,x) + P_2(t,x))$$

Open Questions

Applications:

- How powerful is the primitive with extensions?
- E.g. Euclidian distance is achievable via

$$P(T_X, T_Y, X, Y) = (X - T_X)^2 + (Y - T_Y)^2$$

Applications of CH20:

- Fits many group-based commitment/signature scenarios

$$\mathcal{L}_{\mathsf{alg}} = \{\vec{x} \in \mathbb{G}^l \mid \exists \vec{w} \in \mathbb{Z}_p^t : M(\vec{x}) \cdot \vec{w} = \vec{x}\}$$

- Graph statistics? Asynchronous view?
- Which languages are blinding compatible?

Performance:

- Does [GKLS24] log-size optimisation apply to the updatable case?
- Supporting bigger poly-sized d?

Summary

- New notion: updatable blueprints
  - Regulator sets t, users update x
  - Regulator learns only $P(\sum x_i, t)$
- Efficiency via updatable algebraic NIZK

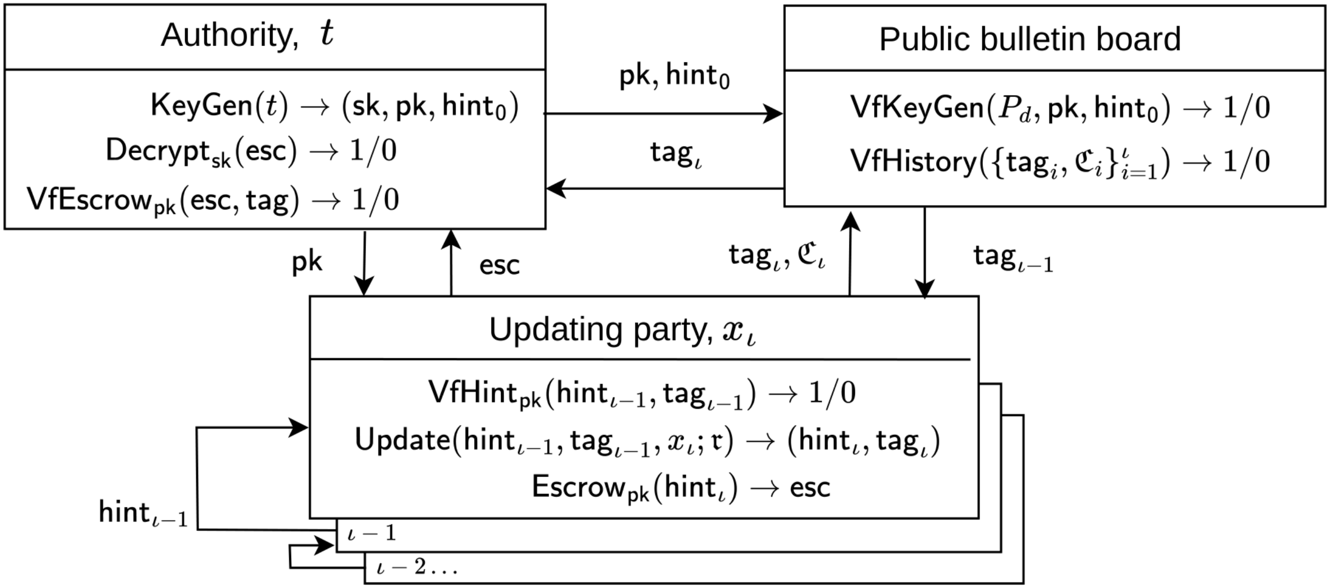
  ([CH20], of independent interest)
- Extendable to more powerful predicates and applications

ia.cr/2023/1787

github.com/volhovm/ublu-impl/

Thank you!

Questions?


Authority, t
$\mathsf{KeyGen}(t) \to (\mathsf{sk}, \mathsf{pk}, \mathsf{hint}_0)$
$\mathsf{Decrypt}_{\mathsf{sk}}(\mathsf{esc}) \to 1/0$
$\mathsf{VfEscrow}_{\mathsf{pk}}(\mathsf{esc}, \mathsf{tag}) \to 1/0$
$\mathsf{pk}, \mathsf{hint}_0$
tag_ι
Public bulletin board
$\mathsf{VfKeyGen}(P_d, \mathsf{pk}, \mathsf{hint}_0) \to 1/0$
$\mathsf{VfHistory}(\{\mathsf{tag}_i, \mathfrak{C}_i\}_{i=1}^{\iota}) \to 1/0$
pk
esc
$\mathsf{tag}_\iota, \mathfrak{C}_\iota$
$\mathsf{tag}_{\iota-1}$
Updating party, x_ι
$\mathsf{VfHint}_{\mathsf{pk}}(\mathsf{hint}_{\iota-1}, \mathsf{tag}_{\iota-1}) \to 1/0$
$\mathsf{Update}(\mathsf{hint}_{\iota-1}, \mathsf{tag}_{\iota-1}, x_\iota; \mathfrak{r}) \to (\mathsf{hint}_\iota, \mathsf{tag}_\iota)$
$\mathsf{Escrow}_{\mathsf{pk}}(\mathsf{hint}_\iota) \to \mathsf{esc}$
$\mathsf{hint}_{\iota-1}$
$\iota - 1$
$\iota - 2 \ldots$

CH20 Updatability: Blinding Compatible Transformations

Issue:

witnesses are not distributed uniformly,
but blinder for commitment stage is!

Therefore: our transformations have to work for pseudo-witnesses too

$$\mathfrak{T} = G^{t}\mathfrak{H}^{r_\mathsf{t}} \qquad \mathfrak{X} = G^{\hat{x}}\mathfrak{H}^{\hat{r}_\mathsf{x}}$$

$$\mathfrak{A} = G^{\alpha}\mathfrak{H}^{r_\alpha} \qquad (A_i, D_i) = (G^{\hat{r}_i}, G^{(\hat{x}-t)^i} H^{\hat{r}_i} W_i^{\alpha}) \text{ for } i \in [d]$$

Describe BC issue and show matrices for original / BC lang

1. $S'_{\mathfrak{T}} = S_{\mathfrak{T}}.$
2. $S'_{\mathfrak{X}} = S_{\mathfrak{X}} G^{U_x} \mathfrak{H}^{U_{r_\times}}.$
3. $S'_{\mathfrak{A}} = G^{U_\alpha} \mathfrak{H}^{U_{r_\alpha}}.$
4. $S'_{A_1} = S_{A_1} G^{U_{r_1}}$
5. $S'_{D_1} = S_{D_1} G^{U_x} H^{U_{r_1}} W_1^{U_\alpha}$
6. $S'_{A_i} = \left(\prod_{j=1}^{i} (S_{A_j})^{V_{i,j}(U_x)}\right) G^{U_{r_i}}$ for $i \in [1, d]$
7. $S'_{D_i} = G^{{U_x}^i} \left(\prod_{j=1}^{i} S_{D_j}^{\binom{i-1}{j-1} U_x^{i-j}}\right)\left(\prod_{j=1}^{i-1} D_j^{\binom{i-1}{j} U_x^{i-j}}\right) H^{U_{r_i}} W_i^{U_\alpha}$ for $i \in [1, d]$
8. $S'_{3+2d+1} = S_{3+2d+1}$
9. $S'_{3+2d+2} = 1$
10. $S'_{5+2d+i} = \left(\prod_{j=1}^{i} (S_{5+2d+j})^{V_{i,j}(U_x)}\right)\left(\prod_{j=1}^{i} (A_j)^{\binom{i}{j} U_x^{i-j+1}}\right)\left(\prod_{j=1}^{i} (S_{A_j})^{-\binom{i}{j} U_x^{i-j+1}}\right)$
    for $i \in [1, d-1]$

$\underline{\mathsf{BlindPowers}(\{B_i\}_{i\in[d]}, \alpha)}$

return $\{D_i := B_i \cdot W_i^{\alpha}\}_{i\in[d]}$

$\underline{\mathsf{Evaluate}(\{A_i, B_i\}_{i\in[d]}, \beta)}$

Let $\{U_i\}_{i=1}^{d}$ be Stirling numbers as defined in Section 5.1.

return $\left(\prod_{i\in[d]} (A_i^{U_i})^{\beta}, \prod_{i\in[d]} (B_i^{U_i})^{\beta}\right)$

$\underline{\mathsf{UpdatePowers}_{\mathsf{pk}}(\{A_{\iota-1,i}, B_{\iota-1,i}\}_{i\in[d]}, x_\iota, \{r_{\iota,i}\}_{i\in[d]})}$

Let $V_{i,j}(X) := \binom{i}{j} X^{i-j}$

for $i \in [d]$ **do**

  $A_{\iota,i} \leftarrow \left(\prod_{j=1}^{i} (A_{\iota-1,j})^{V_{i,j}(x_\iota)}\right) G^{r_{\iota,i}}$

  $B_{\iota,i} \leftarrow \left(G^{x_\iota^i} \prod_{j=1}^{i} (B_{\iota-1,j})^{V_{i,j}(x_\iota)}\right) H^{r_{\iota,i}}$

return $\{A_{\iota,i}, B_{\iota,i}\}_{i\in[d]}$

$\underline{\mathsf{Setup}(1^\lambda, \mathfrak{pp})}$

% To ensure $\mathbb{G}_1$ is the same for Pedersen BCS and the pairing system

parse $\mathfrak{pp}$ as $(\mathbb{G}_1, G, \mathfrak{H}, P_d)$

$\mathsf{pp}_{\mathsf{BLG}} \leftarrow \mathsf{BLG.Setup}(1^\lambda; \mathbb{G}_1, G)$

% Blinding factors for $\{D_i\}_{i=1}^d$

% d comes from the predicate P_d in $\mathfrak{pp}$

$\{W_i\}_{i\in[d]} \xleftarrow{\$} \mathbb{G}_1$

$(\mathsf{crs}_\Pi, \mathsf{td}_\Pi) \leftarrow \Pi.\mathsf{Setup}(1^\lambda, \mathsf{pp}_{\mathsf{BLG}})$

$(\mathsf{crs}_{\Pi_\mathsf{u}}, \mathsf{td}_{\Pi_\mathsf{u}}) \leftarrow \Pi_\mathsf{u}^{\mathcal{L}_\mathsf{c}}.\mathsf{Setup}(1^\lambda, \mathsf{pp}_{\mathsf{BLG}})$

$\mathsf{pp} \leftarrow (\mathfrak{pp}, \mathsf{pp}_{\mathsf{BLG}}, \{W_i\}_{i\in[d]}, 0, P_d, \mathsf{crs}_\Pi, \mathsf{crs}_{\Pi_\mathsf{u}})$

$\mathsf{td} \leftarrow (\mathsf{td}_\Pi, \mathsf{td}_{\Pi_\mathsf{u}})$

return $(\mathsf{pp}, \mathsf{td})$

$\underline{\mathsf{KeyGen}(t)}$

$\mathsf{sk} \xleftarrow{\$} \mathbb{Z}_q,\ H \leftarrow G^{\mathsf{sk}}$

$\{r_{0,i}\}_{i=1}^d, r_\mathsf{t} \xleftarrow{\$} \mathbb{Z}_q$

for $i \in [d]$ **do**

% ElGamal encryptions of t^i

$\quad A_{0,i} \leftarrow G^{r_{0,i}},\ B_{0,i} \leftarrow G^{((-t)^i)} H^{r_{0,i}}$

$\mathfrak{T} \leftarrow \mathfrak{Commit}(t; r_\mathsf{t})$ % Pedersen $\mathfrak{T} = G^t \mathfrak{H}^{r_\mathsf{t}}$

$\mathfrak{X}_0 \leftarrow \mathfrak{Commit}(0; 0)$ % $\mathfrak{X}_0 = 1_{\mathbb{G}_1}$

$\mathfrak{A}_0 \leftarrow \mathfrak{Commit}(0; 0)$

$\mathsf{x_c} \leftarrow (H, \{A_{0,i}, B_{0,i}\}_{i\in[d]}, \mathfrak{T}, \mathfrak{X}_0, \mathfrak{A}_0)$

$\mathsf{w_c} \leftarrow \begin{pmatrix} t, r_\mathsf{t}, \{r_{0,i}\}_{i\in[d]}, \hat{x} := 0, \\ \hat{r}_\mathsf{x} := 0, \{r_{0,i}\cdot(0-t)\}_{i\in[d]}, \\ \alpha := 0, r_\alpha := 0 \\ \alpha\cdot(\hat{x}-t) := 0, r_\alpha(\hat{x}-t) := 0 \end{pmatrix}$

$\pi_\mathsf{c} \xleftarrow{\$} \Pi_\mathsf{u}^{\mathcal{L}_\mathsf{c}}.\mathsf{Prove}(\mathsf{x_c}, \mathsf{w_c})$

$\pi_{\mathsf{pk}} \xleftarrow{\$} \Pi^{\mathcal{L}_{\mathsf{pk}}}.\mathsf{Prove}((H, B_{0,1}, \mathfrak{T}), (\mathsf{sk}, t, r_{0,1}, r_\mathsf{t}))$

$\mathsf{pk} \leftarrow (H, \mathfrak{T}, \pi_{\mathsf{pk}})$

$\mathsf{hint}_0 \leftarrow (\{A_{0,i}, B_{0,i}\}_{i\in[d]}, \mathfrak{X}_0, \pi_\mathsf{c})$

return $(\mathsf{sk}, \mathsf{pk}, \mathsf{hint}_0)$

$\underline{\mathsf{Update}_{\mathsf{pk}}(\mathsf{hint}_{\iota-1}, \mathsf{tag}_{\iota-1}, x_\iota; \mathfrak{r})}$

parse $\mathsf{tag}_{\iota-1}$ as $(\pi_{\mathsf{t},\iota-1}, \mathfrak{X}_{\iota-1})$

$r_{\mathsf{x},\iota} \xleftarrow{\$} \mathbb{Z}_q$

$\mathsf{hint}_\iota \leftarrow \mathsf{UpdateHint}(\mathsf{hint}_{\iota-1}, x_\iota, r_{\mathsf{x},\iota})$

$\mathfrak{C}_\iota \leftarrow \mathfrak{Commit}(x_\iota, \mathfrak{r})$

parse $\mathsf{hint}_{\iota-1}$ as $(\cdot, \mathfrak{X}_{\iota-1}, \cdot)$

parse hint_ι as $(\cdot, \mathfrak{X}_\iota, \cdot)$

$\pi_{\mathsf{t},\iota} \xleftarrow{\$} \Pi^{\mathcal{L}_\mathsf{t}}.\mathsf{Prove}\left(\begin{pmatrix} H, \\ \mathfrak{X}_{\iota-1}, \mathfrak{X}_\iota, \\ \mathfrak{C}_\iota, \pi_{\mathsf{t},\iota-1} \end{pmatrix}, \begin{pmatrix} x_\iota, \\ r_{\mathsf{x},\iota}, \\ \mathfrak{r}_\iota \end{pmatrix}\right)$

$\mathsf{tag}_\iota \leftarrow (\pi_{\mathsf{t},\iota}, \mathfrak{X}_\iota)$

return $(\mathsf{hint}_\iota, \mathsf{tag}_\iota)$

$\underline{\mathsf{UpdateHint}_{\mathsf{pk}}(\mathsf{hint}_{\iota-1}, x_\iota, r_{\mathsf{x},\iota})}$ (Helper)

parse $\mathsf{hint}_{\iota-1}$ as $\begin{pmatrix} \{A_{\iota-1,i}, B_{\iota-1,i}\}_{i\in[d]}, \\ \mathfrak{X}_{\iota-1}, \pi_{\mathsf{c},\iota-1}, \end{pmatrix}$

$\{r_{\iota,i}\}_{i\in[d]} \xleftarrow{\$} \mathbb{Z}_q$

% $r_{\mathsf{x},\iota}$ is the input to UpdateHint

$\mathfrak{X}_\iota \leftarrow \mathfrak{X}_{\iota-1}\cdot\mathfrak{Commit}(x_\iota; r_{\mathsf{x},\iota})$

% $= \mathfrak{Commit}(\sum_{i\in[\iota]} x_i; \sum r_{\mathsf{x},i})$

$\{A_{\iota,i}, B_{\iota,i}\}_{i\in[d]} \leftarrow \mathsf{UpdatePowers}\begin{pmatrix} \{A_{\iota-1,i}, B_{\iota-1,i}\}_{i\in[d]}, \\ x_\iota, \{r_{\iota,i}\}_{i\in[d]} \end{pmatrix}$

$\mathsf{x_c} \leftarrow \begin{pmatrix} H, \{A_{\iota-1,i}, B_{\iota-1,i}\}_{i\in[d]}, \\ \mathfrak{T}, \mathfrak{X}_{\iota-1}, \mathfrak{A} := 1 \end{pmatrix}$

$\mathsf{w_{upd,c}} \leftarrow \begin{pmatrix} x_\iota, \{r_{\iota,i}\}_{i\in[d]}, r_{\mathsf{x},\iota}, \\ \alpha := 0, r_\alpha := 0 \end{pmatrix}$

$\pi_{\mathsf{c},\iota} \xleftarrow{\$} \Pi_\mathsf{u}^{\mathcal{L}_\mathsf{c}}.\mathsf{Update}(\pi_{\mathsf{c},\iota-1}; \mathsf{x_c}, \vec{T}_{\mathsf{upd}}(\mathsf{w_{upd,c}}))$

$\mathsf{hint}_\iota \leftarrow (\{A_{\iota,i}, B_{\iota,i}\}_{i\in[d]}, \mathfrak{X}_\iota, \pi_{\mathsf{c},\iota})$

return hint_ι

$\underline{\mathsf{Escrow}_{\mathsf{pk}}(\mathsf{hint}_\iota)}$

% Partially rerandomize the hint

$(\{A_i, B_i\}_{i\in[d]}, \mathfrak{X}, \pi_\mathsf{c}) \leftarrow \mathsf{UpdateHint}_{\mathsf{pk}}(\mathsf{hint}_\iota, 0, 0)$

$\alpha, \beta, r_\alpha, r_\beta \xleftarrow{\$} \mathbb{Z}_q^\star$

$\mathfrak{A} \leftarrow \mathfrak{Commit}(\alpha; r_\alpha)$

$\mathfrak{B} \leftarrow \mathfrak{Commit}(\beta; r_\beta)$

$\{D_i\}_{i\in[d]} \leftarrow \mathsf{BlindPowers}(\{B_i\}_{i\in[d]}, \alpha)$

$(E_1, E_2) \leftarrow \mathsf{Evaluate}(\{A_i, B_i\}_{i\in[d]}, \beta)$

$\mathsf{x_c} \leftarrow (H, \{A_i, B_i\}_{i\in[d]}, \mathfrak{T}, \mathfrak{X}, \mathfrak{A} := 1)$

$\mathsf{w_{upd,c}} \leftarrow \begin{pmatrix} x_\iota := 0, \{r_{\iota,i} := 0\}_{i\in[d]}, \\ r_{\mathsf{x},\iota} := 0, \alpha, r_\alpha \end{pmatrix}$

$\pi'_\mathsf{c} \xleftarrow{\$} \Pi_\mathsf{u}^{\mathcal{L}_\mathsf{c}}.\mathsf{Update}(\pi_\mathsf{c}; \mathsf{x_c}, T_{\mathsf{upd}}(\mathsf{w_{upd,c}}))$

$\mathsf{w_e} \leftarrow (\alpha, r_\alpha, \beta, r_\beta)$

% U_i are the Stirling numbers

$\pi_\mathsf{e} \leftarrow \Pi^{\mathcal{L}_\mathsf{e}}.\mathsf{Prove}\left(\begin{pmatrix} E_1, E_2, \mathfrak{B}, \mathfrak{A}, \\ \prod A_i^{U_i}, \prod D_i^{U_i} \end{pmatrix}, \mathsf{w_e}\right)$

$\mathsf{esc} \leftarrow \begin{pmatrix} E_1, E_2, \pi_\mathsf{e}, \pi'_\mathsf{c}, \mathfrak{X}, \\ \{A_i, D_i\}_{i\in[d]}, \mathfrak{A}, \mathfrak{B} \end{pmatrix}$

return esc

$\underline{\mathsf{Decrypt}_{\mathsf{sk}}(\mathsf{esc})}$

parse esc as $(E_1, E_2, \cdot)$

$M \leftarrow E_1^{-\mathsf{sk}} * E_2$

% ElGamal decryption $M = G^{\beta P(t,\hat{x})}$

return $[M \stackrel{?}{=} 1_{\mathbb{G}_1}]$

$\mathsf{VfKeyGen}(P_d, \mathsf{pk}, \mathsf{hint}_0)$

parse pk as $(H, \mathfrak{T}, \pi_{\mathsf{pk}})$
parse hint_0 as $\left(\{A_i, B_i\}_{i\in[d]}, \mathfrak{X}, \pi_{\mathsf{c}}\right)$
assert $\mathfrak{X} = 1_{\mathbb{G}}$
assert $\Pi^{\mathcal{L}_{\mathsf{pk}}}.\mathsf{Verify}(\pi_{\mathsf{pk}}; (H, B_1, \mathfrak{T}))$
assert $\Pi_{\mathsf{u}}^{\mathcal{L}_{\mathsf{c}}}.\mathsf{Verify}\left(\pi_{\mathsf{c}}; \begin{pmatrix} \{A_i, B_i\}_{i\in[d]}, \\ \mathfrak{T}, \mathfrak{X}, H, \mathfrak{A} := 1 \end{pmatrix}\right)$
return 1

$\mathsf{VfHistory}_{\mathsf{pk}}(\{\mathsf{tag}_i, \mathfrak{C}_i\}_{i=1}^{\iota})$

Set $\mathfrak{X}_0 \leftarrow 1_{\mathbb{G}}, \pi_{\mathsf{t},0} \leftarrow \pi_{\mathsf{pk}}$
parse tag_ι as $(\pi_{\mathsf{t},\iota}, \mathfrak{X}_\iota)$ for all $i \in [\iota]$
for $i \in [\iota]$ **do**
  assert $\Pi^{\mathcal{L}_{\mathsf{t}}}.\mathsf{Verify}\left(\pi_{\mathsf{t},i}; \begin{pmatrix} H, \mathfrak{X}_{i-1}, \mathfrak{X}_i, \\ \mathfrak{C}_i, \pi_{\mathsf{t},i-1} \end{pmatrix}\right)$
return 1

$\mathsf{VfHint}_{\mathsf{pk}}(\mathsf{hint}, \mathsf{tag})$

parse hint as $\left(\{A_i, B_i\}_{i\in[d]}, \mathfrak{X}, \pi_{\mathsf{c}}\right)$
parse tag as $(\pi_{\mathsf{t}}, \mathfrak{X})$
return $\Pi_{\mathsf{u}}^{\mathcal{L}_{\mathsf{c}}}.\mathsf{Verify}\left(\pi_{\mathsf{c}}; \begin{pmatrix} H, \{A_i, B_i\}_{i\in[d]} \\ \mathfrak{T}, \mathfrak{X}, \mathfrak{A} := 1 \end{pmatrix}\right)$

$\mathsf{VfEscrow}_{\mathsf{pk}}(\mathsf{esc}, \mathsf{tag})$

parse esc as $\begin{pmatrix} E_1, E_2, \pi_{\mathsf{e}}, \pi_{\mathsf{c}}, \mathfrak{X} \\ \{A_i, D_i\}_{i\in[d]}, \mathfrak{A}, \mathfrak{B} \end{pmatrix}$
parse tag as $(\cdot, \mathfrak{X}')$
assert $\mathfrak{X}' = \mathfrak{X}$
assert $\Pi_{\mathsf{u}}^{\mathcal{L}_{\mathsf{c}}}.\mathsf{Verify}\left(\pi_{\mathsf{c}}; \begin{pmatrix} H, \{A_i, D_i\}_{i\in[d]}, \\ \mathfrak{T}, \mathfrak{X}, \mathfrak{A} \end{pmatrix}\right)$
assert $\Pi^{\mathcal{L}_{\mathsf{e}}}.\mathsf{Verify}\left(\pi_{\mathsf{e}}; \begin{pmatrix} H, E_1, E_2, \mathfrak{B}, \\ \mathfrak{A}, \{A_i, D_i\}_{i\in[d]} \end{pmatrix}\right)$
return 1

Definition 7 (Correctness). *Let* BC *and* $P \in \mathcal{P}_{\mathbb{V}}$ *be as in Definition 6, and* $\lambda \in \mathbb{N}$*. A UPPB scheme for* (BC, P) *is* correct *if the following statements hold for all* $\mathfrak{pp} \xleftarrow{\$} \mathfrak{Setup}(1^\lambda)$*,* $(\mathsf{pp}, \cdot) \xleftarrow{\$} \mathsf{Setup}(1^\lambda, \mathfrak{pp})$ *(remember these are implicit in all the algorithms):*

– *Full correctness: for all* $t \in \mathbb{V}$*, all poly-sized sequences of values* $x_1, \ldots, x_n \in \mathbb{V}$ *and* $\mathfrak{r}_1, \ldots, \mathfrak{r}_n \in \mathbb{R}$*:*

$$\Pr\left[\begin{array}{l} \mathsf{VfKeyGen}(\mathsf{pk}, \mathsf{hint}_0) = 1 \wedge \\ \text{for all } i \in [n]: \\ \quad \mathsf{VfHint}_{\mathsf{pk}}(\mathsf{hint}_i, \mathsf{tag}_i) = 1 \wedge \\ \quad \mathsf{VfHistory}_{\mathsf{pk}}(\{\mathsf{tag}_j, \mathfrak{C}_j\}_{j=1}^{i}) = 1 \wedge \\ \quad \mathsf{VfEscrow}_{\mathsf{pk}}(\mathsf{esc}_i, \mathsf{tag}_i) = 1 \wedge \\ \quad \mathsf{Decrypt}_{\mathsf{sk}}(\mathsf{esc}_i) = P(t, \sum_{j=1}^{i} x_j) \end{array} : \begin{array}{l} (\mathsf{sk}, \mathsf{pk}, \mathsf{hint}_0) \xleftarrow{\$} \mathsf{KeyGen}(t) \\ \text{for } i \in [n]: \\ \quad (\mathsf{hint}_i, \mathsf{tag}_i) \Lsh_{\$} \\ \quad\quad \mathsf{Update}_{\mathsf{pk}}(\mathsf{hint}_{i-1}, \mathsf{tag}_{i-1}, x_i, \mathfrak{r}_i) \\ \quad \mathsf{esc}_i \xleftarrow{\$} \mathsf{Escrow}_{\mathsf{pk}}(\mathsf{hint}_i) \\ \quad \mathfrak{C}_i \leftarrow \mathfrak{Commit}(x_i; \mathfrak{r}_i) \end{array}\right] = 1$$

– *Update correctness: for all* $\mathsf{pk}, \mathsf{hint}_0$ *s.t.* $\mathsf{VfKeyGen}(\mathsf{pk}, \mathsf{hint}_0) = 1$*, and all* $\mathsf{hint}_n, \{\mathsf{tag}_j, \mathfrak{C}_j\}_{i=1}^{n}$ *such that* $\mathsf{VfHint}_{\mathsf{pk}}(\mathsf{hint}_n, \mathsf{tag}_n) = 1$ *and*
$\mathsf{VfHistory}_{\mathsf{pk}}(\{\mathsf{tag}_j, \mathfrak{C}_j\}_{j=1}^{n}) = 1$*, and for all* $x \in \mathbb{V}, \mathfrak{r} \in \mathbb{R}$*:*

$$\Pr\left[\begin{array}{l} \mathsf{VfHint}_{\mathsf{pk}}(\mathsf{hint}_{n+1}, \mathsf{tag}_{n+1}) = 1 \wedge \\ \mathsf{VfHistory}_{\mathsf{pk}}(\{\mathsf{tag}_j, \mathfrak{C}_j\}_{j=1}^{n+1}) = 1 \wedge \\ \mathsf{VfEscrow}_{\mathsf{pk}}(\mathsf{esc}_{n+1}, \mathsf{tag}_{n+1}) = 1 \end{array} : \begin{array}{l} (\mathsf{hint}_{n+1}, \mathsf{tag}_{n+1}) \Lsh_{\$} \\ \quad \mathsf{Update}_{\mathsf{pk}}(\mathsf{hint}_n, \mathsf{tag}_n, x, \mathfrak{r}) \\ \mathsf{esc}_{n+1} \xleftarrow{\$} \mathsf{Escrow}_{\mathsf{pk}}(\mathsf{hint}_{n+1}) \\ \mathfrak{C}_{n+1} \leftarrow \mathfrak{Commit}(x; \mathfrak{r}) \end{array}\right] = 1$$

In both statements, the probability is taken over the random coins internally sampled by the randomized algorithms of UPPB.

Definition 9 (Soundness). *A UPPB scheme for* (BC, P) *is* sound *if there exists a deterministic poly-time black-box extractor* Ext*, such that for all PPT* $\mathcal{A}$*:*

1. *Valid history can be explained in terms of base commitments: for all* $\iota > 0$,

$$\Pr\left[\begin{array}{l} \mathsf{VfKeyGen}(\mathsf{pk}, \mathsf{hint}_0) = 1 \wedge \\ \mathsf{VfHistory}_{\mathsf{pk}}(\{\mathsf{tag}_i, \mathfrak{C}_i\}_{i=1}^{\iota}) = 1 \wedge \\ \mathfrak{C}_\iota \neq \mathfrak{Commit}(x_\iota, \mathfrak{r}_\iota) \end{array} : \begin{array}{l} \mathfrak{pp} \xleftarrow{\$} \mathfrak{Setup}(1^\lambda) \\ (\mathsf{pp}, \mathsf{td}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda, \mathfrak{pp}) \\ (\mathsf{pk}, \mathsf{hint}_0, \{\mathsf{tag}_i, \mathfrak{C}_i\}_{i=1}^{\iota}) \xleftarrow{\$} \\ \hfill \mathcal{A}(\mathsf{pp}) \\ (x_\iota, \mathfrak{r}_\iota) \leftarrow \mathsf{Ext}(\mathsf{td}, \mathsf{tag}_\iota) \end{array}\right] \leq \mathsf{negl}(\lambda)$$

2. *Decryption always reveals the predicate computed for the sum of update values: for all* $t \in \mathbb{V}$, $\iota > 0$,

$$\Pr\left[\begin{array}{l} \mathsf{VfHistory}_{\mathsf{pk}}(\{\mathsf{tag}_i, \mathfrak{C}_i\}_{i=1}^{\iota}) = 1 \wedge \\ \mathsf{VfEscrow}_{\mathsf{pk}}(\mathsf{esc}^\star, \mathsf{tag}_\iota) = 1 \wedge \\ \mathsf{Decrypt}_{\mathsf{sk}}(\mathsf{esc}^\star) \neq P(t, \sum_{i=1}^{\iota} x_i) \end{array} : \begin{array}{l} \mathfrak{pp} \xleftarrow{\$} \mathfrak{Setup}(1^\lambda) \\ (\mathsf{pp}, \mathsf{td}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda, \mathfrak{pp}) \\ (\mathsf{sk}, \mathsf{pk}, \mathsf{hint}_0) \xleftarrow{\$} \mathsf{KeyGen}(t) \\ (\mathsf{esc}^\star, \{\mathsf{tag}_i, \mathfrak{C}_i\}_{i=1}^{\iota}) \xleftarrow{\$} \\ \hfill \mathcal{A}(\mathsf{pp}, \mathsf{pk}, \mathsf{hint}_0) \\ \mathsf{for}\ i \in [1, \iota]: \\ \quad (x_i, \mathfrak{r}_i) \leftarrow \mathsf{Ext}(\mathsf{td}, \mathsf{tag}_i) \end{array}\right] \leq \mathsf{negl}(\lambda)$$

Definition 8 (History Binding). *A UPPB scheme for* (BC, P) *is* history binding *if for all PPT* $\mathcal{A}$*, it holds that* $\Pr[\mathcal{G}_{\mathcal{A}}(1^\lambda) = 1] \le \mathsf{negl}(\lambda)$*, where game* $\mathcal{G}_{\mathcal{A}}(1^\lambda)$ *is as follows:*

$$
\begin{array}{ll}
1: & \mathfrak{pp} \xleftarrow{\$} \mathfrak{Setup}(1^\lambda);\ (\mathsf{pp}, \cdot) \xleftarrow{\$} \mathsf{Setup}(1^\lambda, \mathfrak{pp}) \\
2: & (\mathsf{pk}, \mathsf{hint}_0, \{\{\mathsf{tag}_i^{(b)}, \mathfrak{C}_i^{(b)}\}_{i=1}^{\iota}\}_{b\in\{0,1\}}) \xleftarrow{\$} \mathcal{A}(\mathsf{pp}) \\
3: & \textbf{return } \mathsf{VfKeyGen}(\mathsf{pk}, \mathsf{hint}_0) = 1 \wedge \\
4: & \qquad \mathsf{VfHistory}_{\mathsf{pk}}(\{\mathsf{tag}_i^{(0)}, \mathfrak{C}_i^{(0)}\}_{i=1}^{\iota}) = 1 \wedge \\
5: & \qquad \Big(\mathsf{VfHistory}_{\mathsf{pk}}(\{\mathsf{tag}_i^{(0)}, \mathfrak{C}_i^{(0)}\}_{i=1}^{\iota-1}) \neq 1 \vee \\
6: & \qquad\quad \mathsf{VfHistory}_{\mathsf{pk}}(\{\mathsf{tag}_i^{(1)}, \mathfrak{C}_i^{(1)}\}_{i=1}^{\iota}) = 1 \wedge \\
7: & \qquad\quad \mathsf{tag}_\iota^{(0)} = \mathsf{tag}_\iota^{(1)} \wedge \exists i.\, (\mathsf{tag}_i^{(0)}, \mathfrak{C}_i^{(0)}) \neq (\mathsf{tag}_i^{(1)}, \mathfrak{C}_i^{(1)})\Big)
\end{array}
$$

Definition 10 (Threshold Hiding). *A UPPB scheme for* (BC, P) *is* threshold hiding *if for all PPT* $\mathcal{A}$ *it holds that:*

$$\Pr\left[b \stackrel{?}{=} b^\star \; : \; \begin{array}{l} \mathfrak{pp} \xleftarrow{\$} \mathfrak{Setup}(1^\lambda); \; (\mathsf{pp}, \cdot) \xleftarrow{\$} \mathsf{Setup}(1^\lambda, \mathfrak{pp}) \\ (t_0, t_1) \leftarrow \mathcal{A}(\mathsf{pp}), \; b \xleftarrow{\$} \{0,1\} \\ (\cdot, \mathsf{pk}, \mathsf{hint}_0) \xleftarrow{\$} \mathsf{KeyGen}(t_b) \\ b^\star \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{hint}_0) \end{array} \right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

Tag hiding states that tags do not reveal any additional information than already revealed by $\mathfrak{C}$ itself.

Definition 11 (Tag Hiding). *A UPPB scheme for* BC *defined over* $(\mathbb{V}, \mathbb{R})$ *and* $P(T, X) \in \mathcal{P}_{\mathbb{V}}$ *is (perfectly)* hiding in tags *if, for* $(\mathsf{pp}, \mathsf{td}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda, \mathbb{V}, \mathcal{P})$ *all* $t \in \mathbb{V}$, *all* pk, *all pairs* $(\mathsf{hint}, \mathsf{tag})$ *such that* $\mathsf{VfHint}_{\mathsf{pk}}(\mathsf{hint}, \mathsf{tag}) = 1$, *and for all* $x \in \mathbb{V}$, $\mathfrak{r} \in \mathbb{R}$, *there exists a PPT* $\mathcal{S}$ *such that:*

$$\left\{ \mathsf{tag}' \mid (\cdot, \mathsf{tag}') \xleftarrow{\$} \mathsf{Update}_{\mathsf{pk}}(\mathsf{hint}, \mathsf{tag}, x, \mathfrak{r}) \right\} = \left\{ \mathcal{S}(\mathsf{td}, \mathsf{pk}, \mathsf{tag}, \mathfrak{C} := \mathfrak{Commit}(x, \mathfrak{r})) \right\}$$

where distributions are over the internal randomness of the Update *algorithm and the simulator. For the first update, this holds conditioned on* $\mathsf{hint} := \mathsf{hint}_0$, $\mathsf{tag} := \perp$.

Definition 12 (Hint Hiding). *A UPPB scheme for* BC *defined over* $(\mathbb{V}, \mathbb{R})$ *and* $P(T, X) \in \mathcal{P}_{\mathbb{V}}$ *is* (computationally value) hiding in hints *if, for all* $t \in \mathbb{V}$ *and all PPT* $\mathcal{A}$*, it holds that* $\Pr[\mathcal{G}_{\mathcal{A}}(1^\lambda) = 1] \leq 1/2 + \mathsf{negl}(\lambda)$*, where game* $\mathcal{G}_{\mathcal{A}}(1^\lambda)$ *is:*

$\mathfrak{pp} \xleftarrow{\$} \mathfrak{Setup}(1^\lambda)$; $(\mathsf{pp}, \cdot) \xleftarrow{\$} \mathsf{Setup}(1^\lambda, \mathfrak{pp})$

$(\cdot, \mathsf{pk}, \mathsf{hint}_0) \xleftarrow{\$} \mathsf{KeyGen}(t)$; $b \xleftarrow{\$} \{0, 1\}$

$(\mathsf{hint}^\star, \mathsf{tag}^\star, x^{(0)}, x^{(1)}, \mathfrak{r}) \xleftarrow{\$} \mathcal{A}(\mathsf{pp}, \mathsf{pk}, \mathsf{hint}_0)$

$(\mathsf{hint}, \cdot) \xleftarrow{\$} \mathsf{Update}_{\mathsf{pk}}(\mathsf{hint}^\star, \mathsf{tag}^\star, x^{(b)}, \mathfrak{r})$

$b^\star \xleftarrow{\$} \mathcal{A}(\mathsf{hint})$

return $b^\star \stackrel{?}{=} b \wedge \mathsf{VfHint}_{\mathsf{pk}}(\mathsf{hint}^\star, \mathsf{tag}^\star) \stackrel{?}{=} 1$

Definition 13 (Escrow hiding). *A UPPB scheme for* BC *defined over* $(\mathbb{V}, \mathbb{R})$ *and* $P(T, X) \in \mathcal{P}_{\mathbb{V}}$ *is* escrow hiding *if there exists a PPT simulator* $\mathcal{S}$ *such that for all PPT* $\mathcal{A}$*, it holds that* $\Pr[\mathcal{G}_{\mathcal{A}}(1^\lambda) = 1] \leq 1/2 + \mathsf{negl}(\lambda)$*, where game* $\mathcal{G}_{\mathcal{A}}(1^\lambda)$ *is as follows:*

$(\mathsf{pp}, \mathsf{td}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda, \mathbb{V}, \mathcal{P})$.

$(t, \mathsf{pk}, \mathsf{hint}_0, \{\mathsf{tag}_i, x_i, \mathfrak{r}_i\}_{i=1}^{\iota}, \mathsf{hint}_\iota) \xleftarrow{\$} \mathcal{A}(\mathsf{pp})$

$b \xleftarrow{\$} \{0, 1\}$

$\mathsf{esc} \leftarrow$ **if** $b = 0$ **then** $\mathsf{Escrow}_{\mathsf{pk}}(\mathsf{hint}_\iota)$ **else** $\mathcal{S}(\mathsf{td}, \mathsf{pk}, P(t, \sum_{i \in [\iota]} x_i), \mathsf{tag}_\iota)$

$b^\star \xleftarrow{\$} \mathcal{A}(\mathsf{esc})$

return $b^\star = b \wedge$

$\quad \mathsf{VfKeyGen}(\mathsf{pk}, \mathsf{hint}_0) = 1 \wedge$

$\quad \mathsf{VfHistory}_{\mathsf{pk}}(\{\mathsf{tag}_i, \mathfrak{Commit}(x_i, \mathfrak{r}_i)\}_{i=1}^{\iota}) = 1 \wedge$

$\quad \mathsf{VfHint}(\mathsf{hint}_\iota, \mathsf{tag}_\iota) = 1$