Multi-Cloud Storage Management

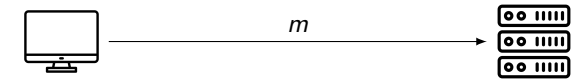
Adam Oumar Abdel-Rahman, Gael Marcadet
Sofiane Azogagh, Zelma Aubin Birba, Arthur Tran Van
Supervised by : Gilles Seghaier

November 2023

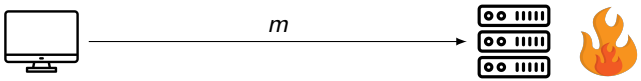
Astran

- Sensitive data storage
- Availability
- 20 employees
- 4.5 M€

Problematic of the Single-storage



Problematic of the Single-storage



User $\mathcal{U}(m)$ Storage $\mathcal{S}(m)$

Can we do better ? YES

- With redundancy, but not fun !
- With **secret sharing**

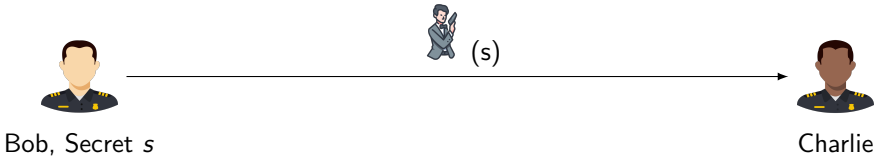
Secret-Sharing (SS)

How to send s ?

Bob, Secret s

Charlie

Secret-Sharing (SS)



Secret-Sharing (SS)

Secret leaked if spy is captured !

Secret-Sharing (SS)

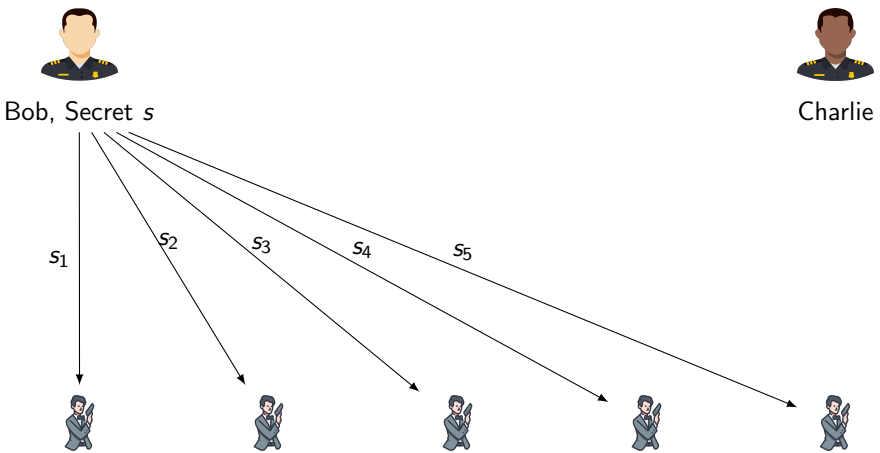
$s = s_1 || \ldots, || s_5$

Bob, Secret s

Charlie

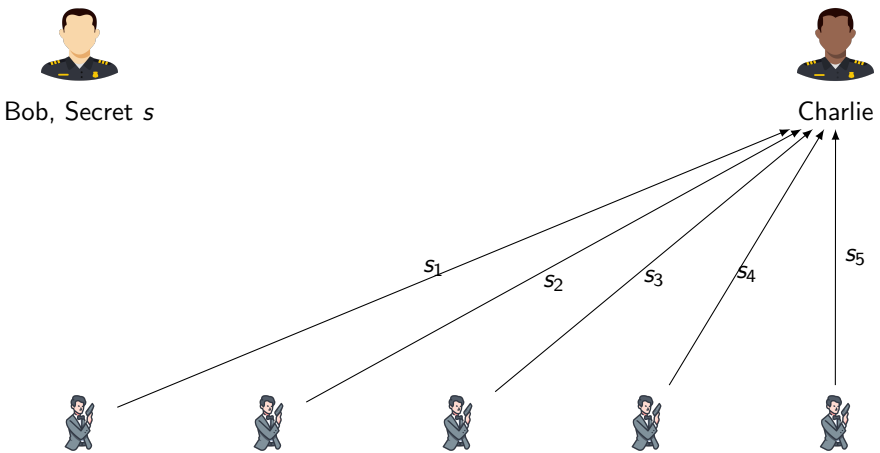
Secret-Sharing (SS)

$s = s_1 || \ldots, || s_5$



Secret-Sharing (SS)

$s = s_1 || \ldots, || s_5$



Secret-Sharing (SS)

$s = s_1 || \dots || s_5$

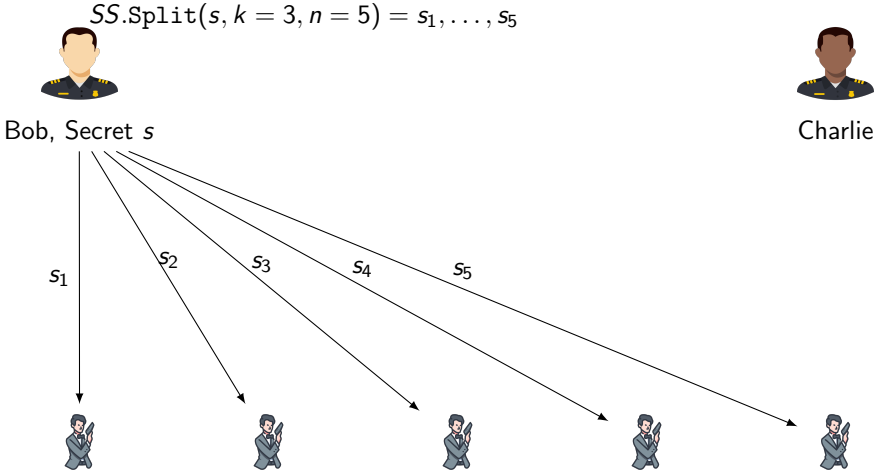
Bob, Secret s

Secret lost if a spy is killed !

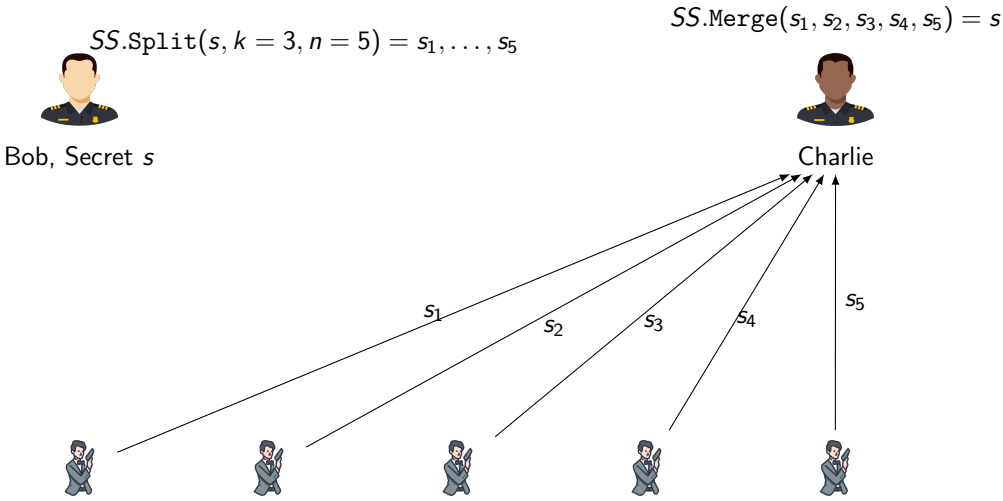
Charlie

s_2 s_3 s_4 s_5

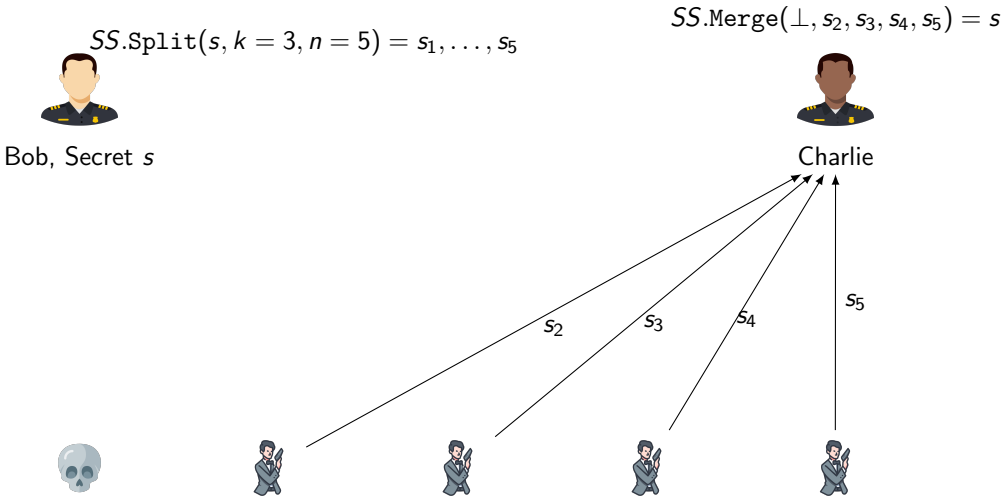
Secret-Sharing (SS)



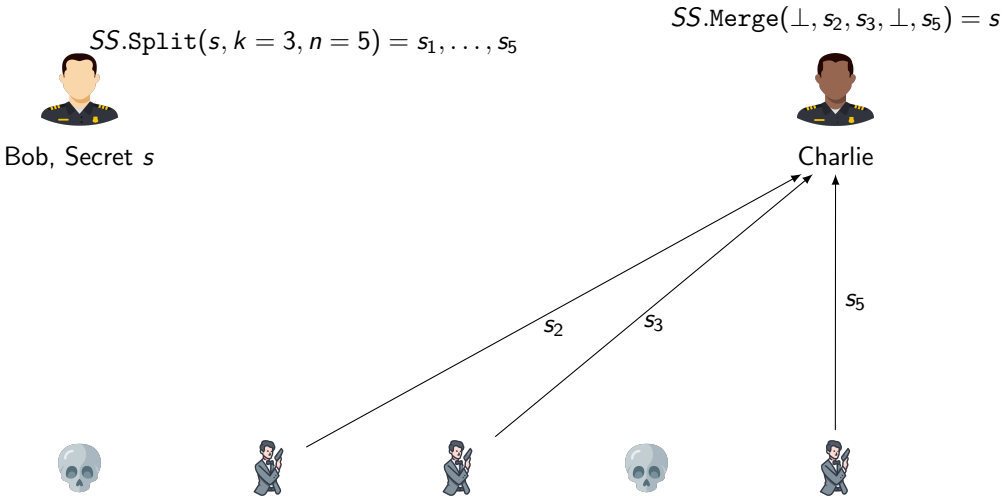
Secret-Sharing (SS)



Secret-Sharing (SS)



Secret-Sharing (SS)



Secret-Sharing (SS)

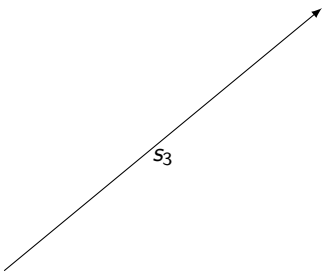
$SS.\texttt{Split}(s, k = 3, n = 5) = s_1, \ldots, s_5$

Number of shares upper than $k - 1 \implies$ Preserved secret!

$SS.\texttt{Merge}(\perp, \perp, s_3, \perp, s_5) = \perp$

Bob, Secret s

Charlie



What's the problem ?

User $\mathcal{U}(m)$

Proxy

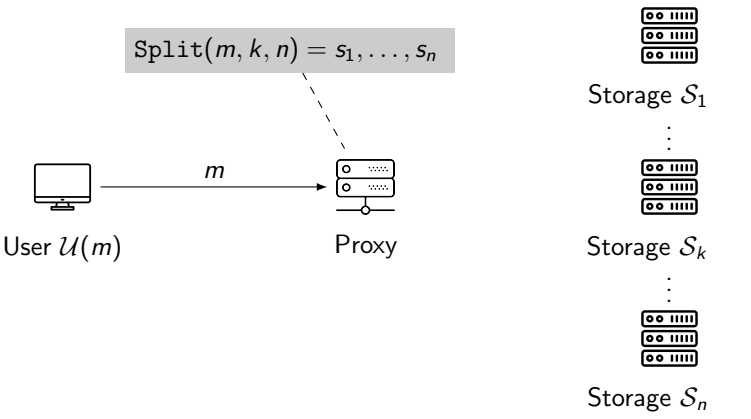
Storage $\mathcal{S}_1$

Storage $\mathcal{S}_k$

Storage $\mathcal{S}_n$

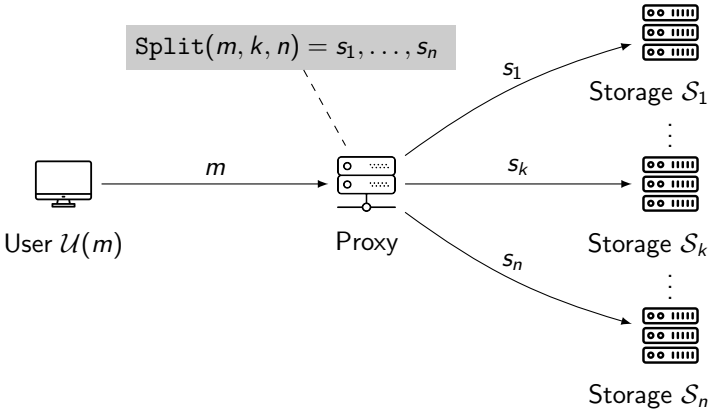
What's the problem ?

Upload



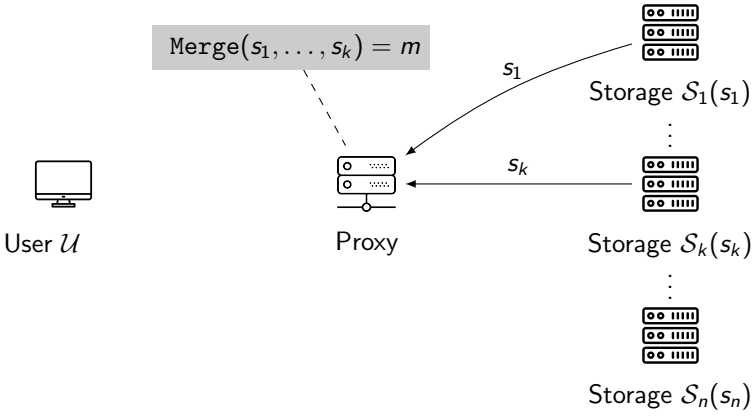
What's the problem ?

Upload



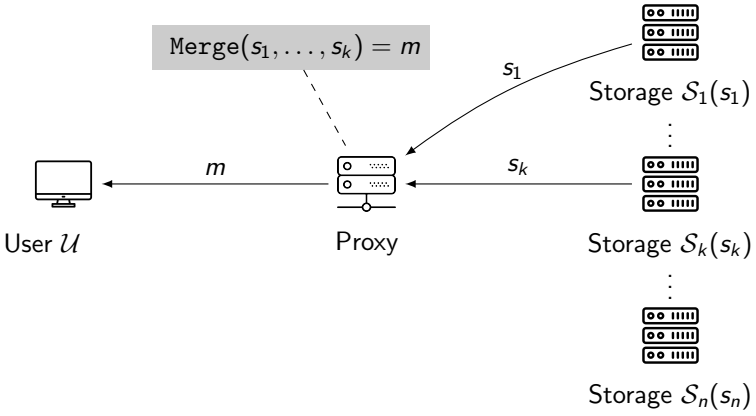
What's the problem ?

Download

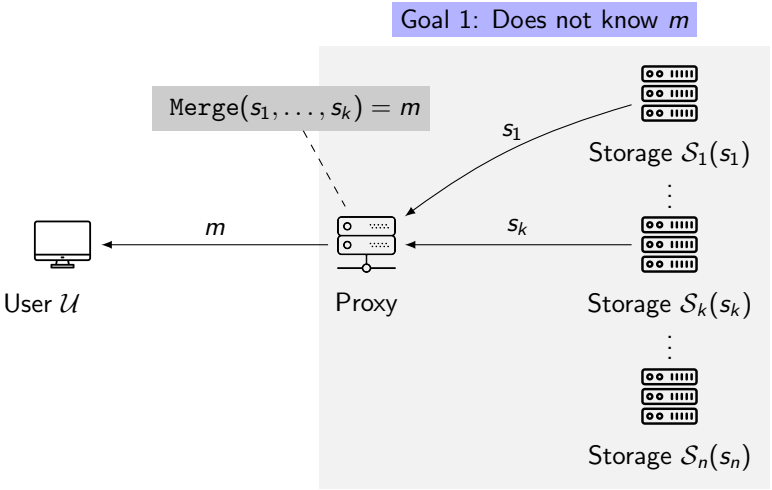


What's the problem ?

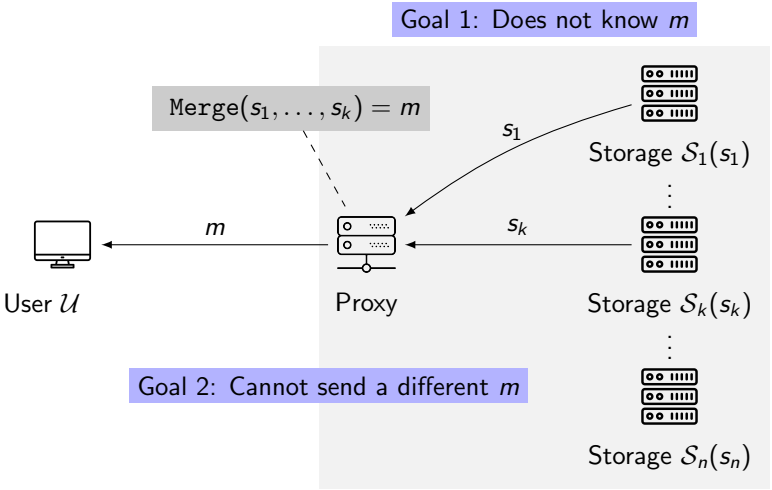
Download



What's the problem ?



What's the problem ?

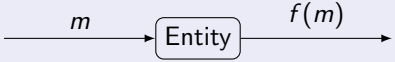


Looking for security, but against what ?

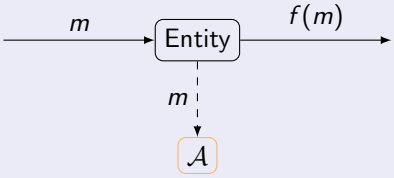
What the adversary (denoted $\mathcal{A}$) is able to corrupt ?

Type of adversary $\mathcal{A}$

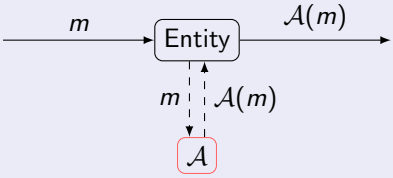
Honest



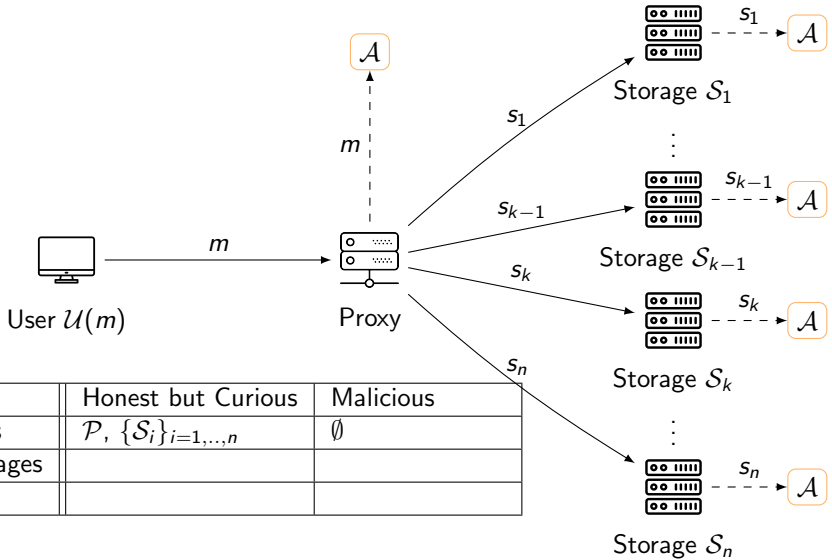
Honest-but-Curious



Malicious

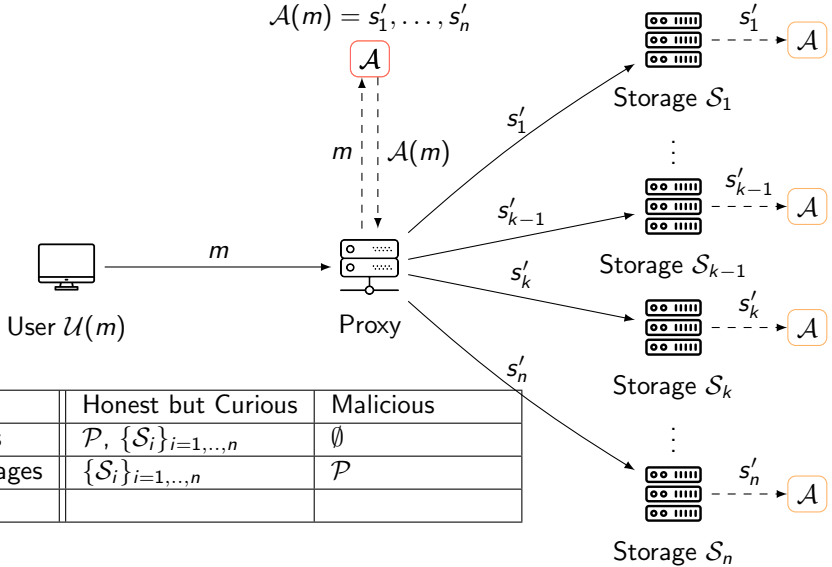


Corruption Model



Corr. Model	Honest but Curious	Malicious
Fully-Curious	$\mathcal{P}$, $\{\mathcal{S}_i\}_{i=1,..,n}$	$\emptyset$
Curious Storages		
Collusion		

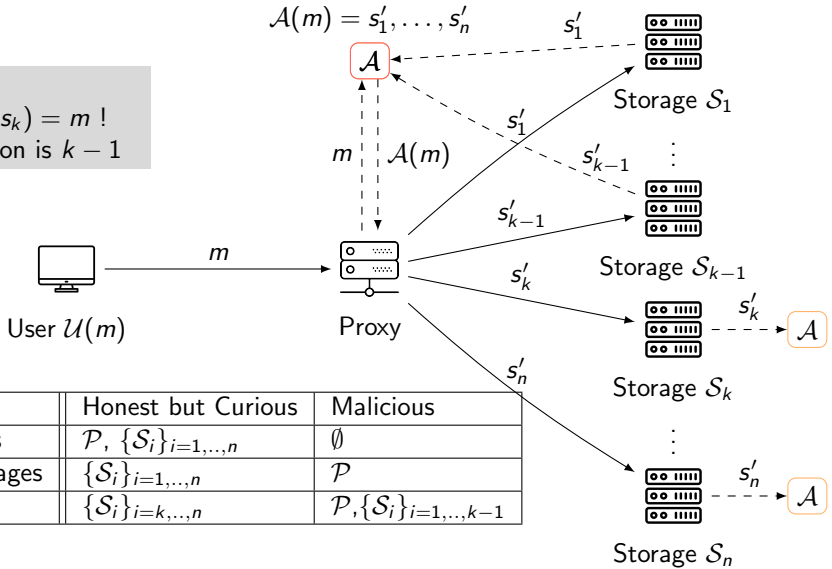
Corruption Model



Corr. Model	Honest but Curious	Malicious
Fully-Curious	$\mathcal{P}$, $\{\mathcal{S}_i\}_{i=1,..,n}$	$\emptyset$
Curious Storages	$\{\mathcal{S}_i\}_{i=1,..,n}$	$\mathcal{P}$
Collusion		

Corruption Model

Recall that
$\texttt{Merge}(s_1, \ldots, s_k) = m$!
So max collusion is $k - 1$



Corr. Model	Honest but Curious	Malicious
Fully-Curious	$\mathcal{P}$, $\{\mathcal{S}_i\}_{i=1,..,n}$	$\emptyset$
Curious Storages	$\{\mathcal{S}_i\}_{i=1,..,n}$	$\mathcal{P}$
Collusion	$\{\mathcal{S}_i\}_{i=k,..,n}$	$\mathcal{P}$,$\{\mathcal{S}_i\}_{i=1,..,k-1}$

Contributions

- 1 Attack against secrecy

Contributions

- 1 Attack against secrecy
- 1 Attack against confidentiality

Contributions

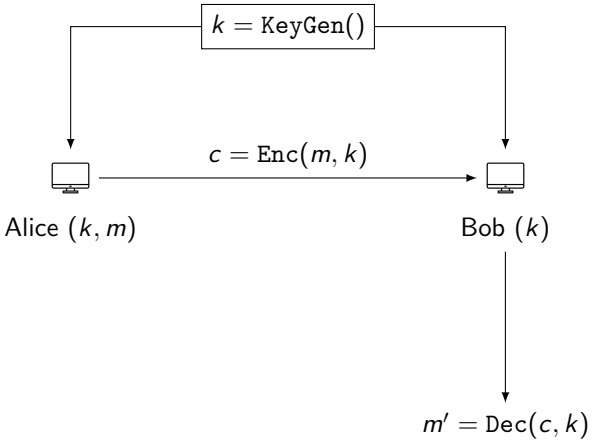
- 1 Attack against secrecy
- 1 Attack against confidentiality
- 3 new protocols

$\mathcal{PRZT}$: The initial (flawed) protocol

scheme	#Users	Proxy model	#Collusions	Secrecy	Integrity
$\mathcal{PRZT}$	1	Honest but Curious	0	✓	✓

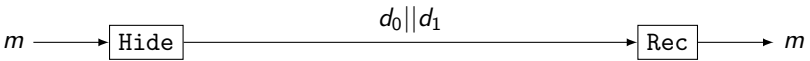
Primitive	Introduced	Algorithms
Secret Sharing	✓	`Split`, `Merge`
Proxy Homomorphic Re-Encryption		`KeyGen`, `Enc`, `Dec`, `ReEnc`
AONT		`Hide`, `Rec`

Symmetric encryption



AONT

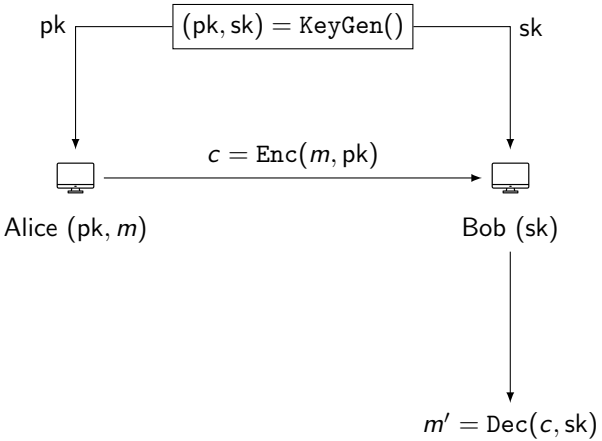
All-or-Nothing



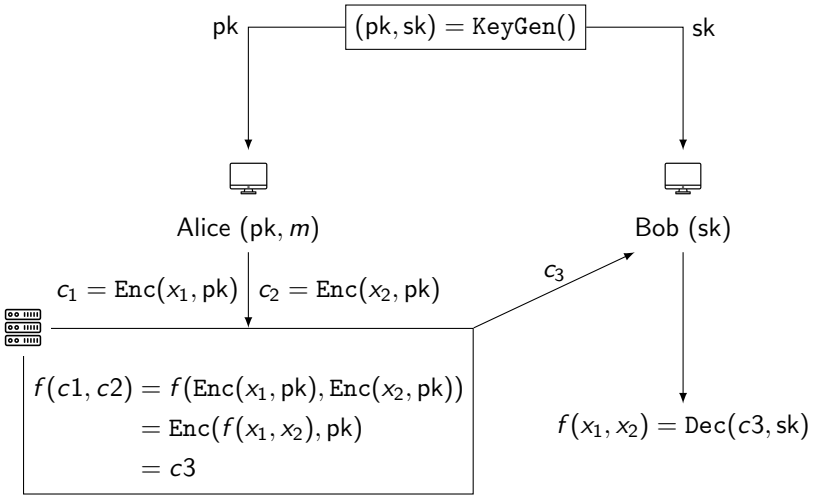
1. $k \leftarrow Sym.\texttt{KeyGen}()$
2. $c \leftarrow Sym.\texttt{Enc}(k, m||0^\lambda)$
3. $d_0 \leftarrow c$
4. $d_1 \leftarrow k \oplus H(c)$

1. $k \leftarrow d_1 \oplus H(d_0)$
2. $m' \leftarrow Sym.\texttt{Dec}(k, d_0)$
3. Check that $m' = m||0^\lambda$

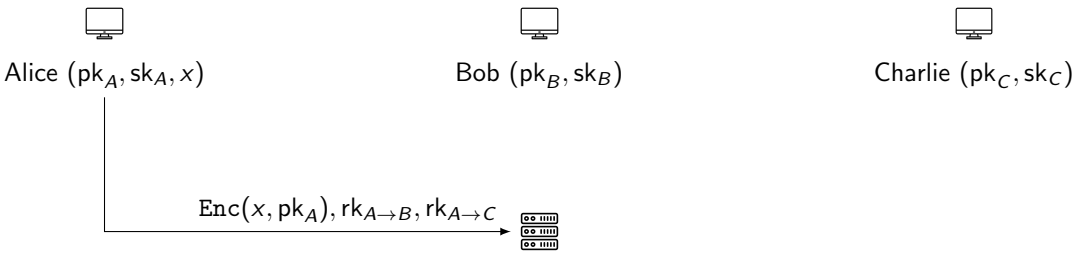
Public Key Encryption (PKE)



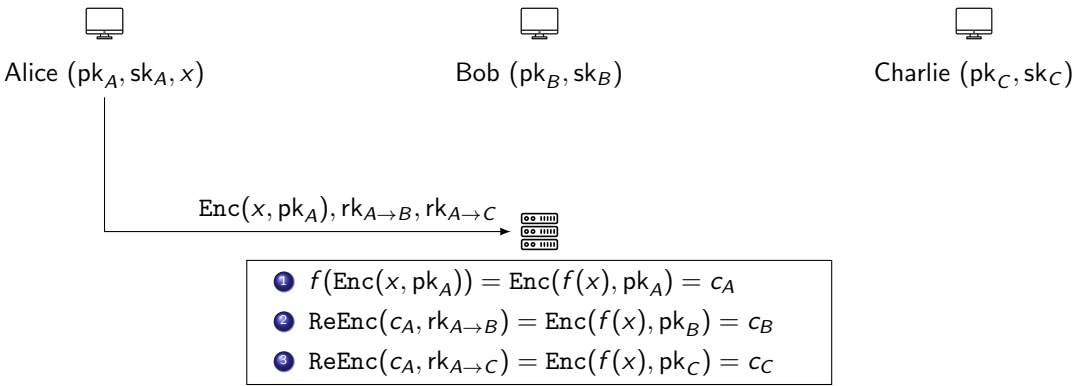
Public Key **Homomorphic** Encryption



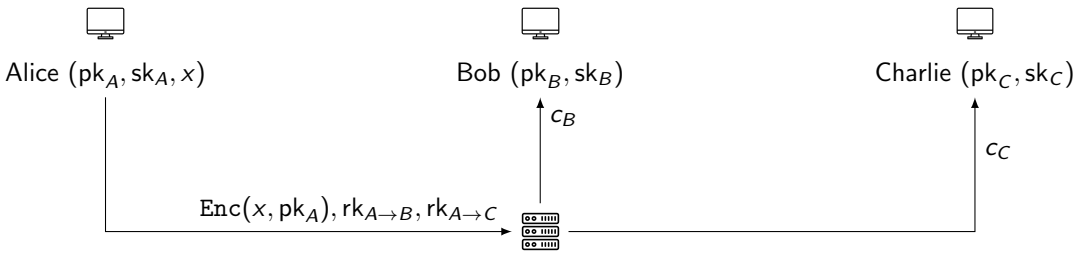
Proxy Homomorphic **Re-Encryption** (PRE)



Proxy Homomorphic **Re-Encryption** (PRE)



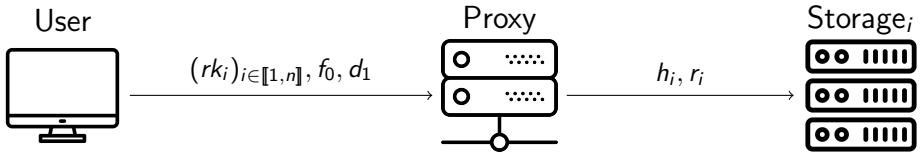
Proxy Homomorphic **Re-Encryption** (PRE)



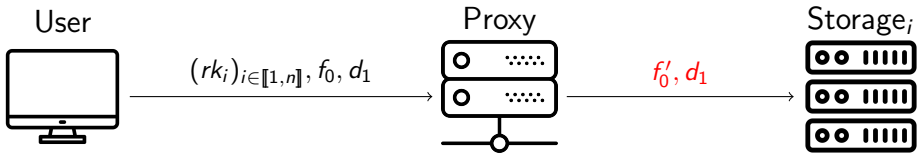
1. $f(\mathtt{Enc}(x, \mathsf{pk}_A)) = \mathtt{Enc}(f(x), \mathsf{pk}_A) = c_A$
2. $\mathtt{ReEnc}(c_A, \mathsf{rk}_{A\to B}) = \mathtt{Enc}(f(x), \mathsf{pk}_B) = c_B$
3. $\mathtt{ReEnc}(c_A, \mathsf{rk}_{A\to C}) = \mathtt{Enc}(f(x), \mathsf{pk}_C) = c_C$

$\mathcal{PRZT}$: The initial (flawed) protocol

Upload only



$\mathcal{PRZT}$: A Forward-Attack



$$d_0||d_1 \leftarrow AONT.\mathtt{Hide}(m)$$
$$\forall i \in [\![1,n]\!], \mathsf{rk}_i \leftarrow PRE.\mathtt{ReEnc}(\mathsf{sk}_u, \mathsf{pk}_i)$$
$$f_0 \leftarrow PRE.\mathtt{Enc}(d_0, \mathsf{sk}_u)$$

$$f'_0 \leftarrow PRE.\mathtt{ReEnc}(\mathsf{rk}_i, f_0)$$

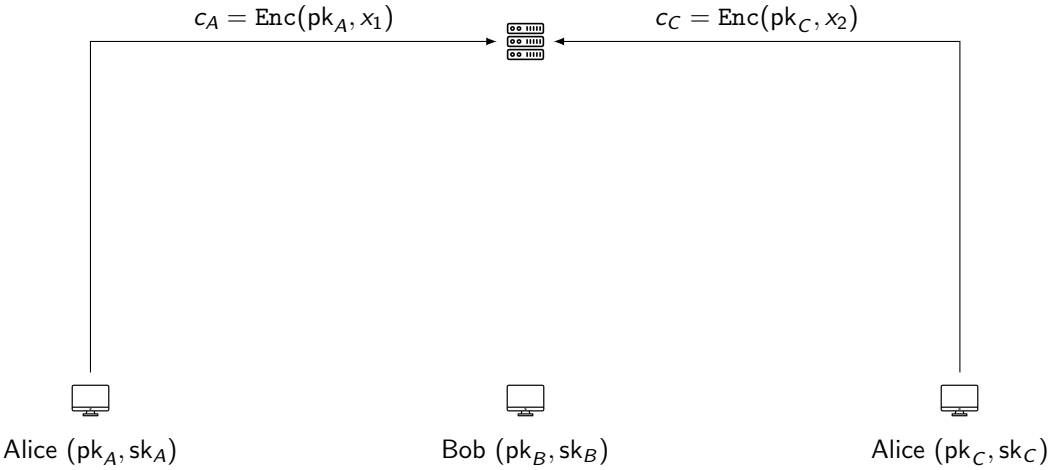
$$d_0 \leftarrow PRE.\mathtt{Dec}(\mathsf{sk}_i, f'_0)$$

$\mathcal{MKZT}$: Multi-Key Homomorphic Encryption-based Protocol

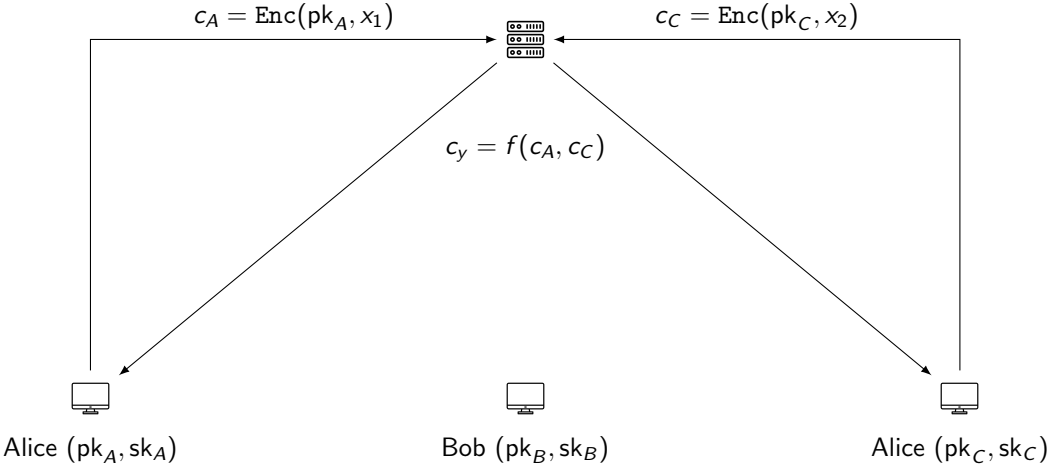
scheme	#Users	Proxy	#Collusions	Secrecy	Integrity
$\mathcal{PRZT}$	1	Honest but Curious	✓	✓	✓
$\mathcal{MKZT}$	1	Honest but Curious	✓	✓	✓

Primitive	Introduced	Algorithms
Secret Sharing (SS)	✓	`Split`, `Merge`
Multi-key Homomorphic Encryption (MKE)		`KeyGen`, `Enc`, `PartDec`, `FinDec`
AONT	✓	`Hide`, `Rec`

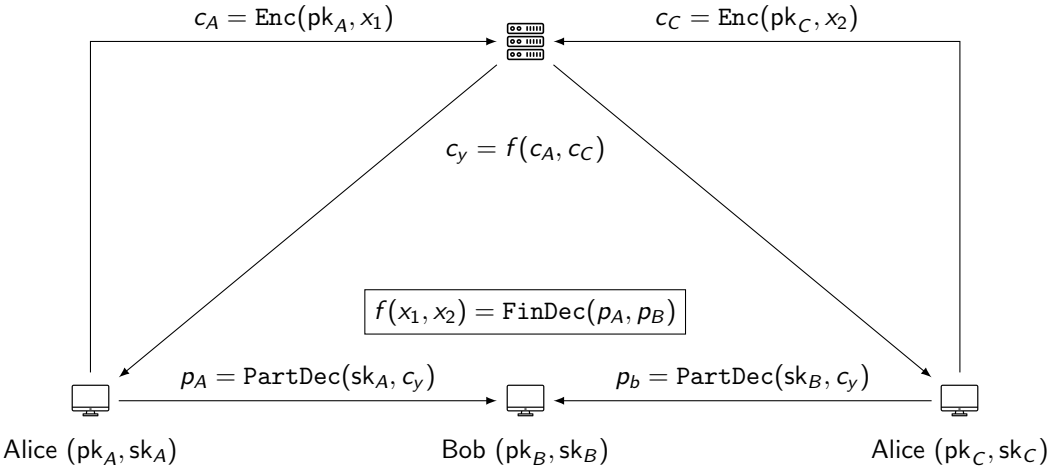
Multi-Key Homomorphic Encryption (MKE)



Multi-Key Homomorphic Encryption (MKE)

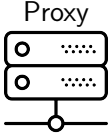


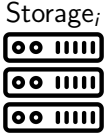
Multi-Key Homomorphic Encryption (MKE)



$\mathcal{MKZT}$

Upload



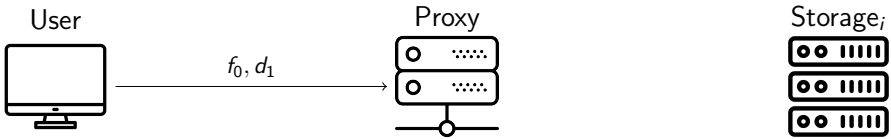


$d_0||d_1 \leftarrow \text{AONT.}\texttt{Hide}(m)$

$f_0 \leftarrow \text{MKE.}\texttt{Enc}((pk_i)_{i\in[\![1,n]\!]}, d_0)$

$\mathcal{MKZT}$

Upload

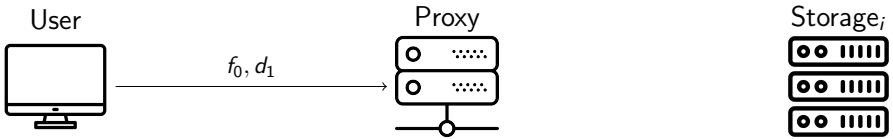


$d_0||d_1 \leftarrow \text{AONT.}\texttt{Hide}(m)$

$f_0 \leftarrow \text{MKE.}\texttt{Enc}((pk_i)_{i\in[\![1,n]\!]}, d_0)$

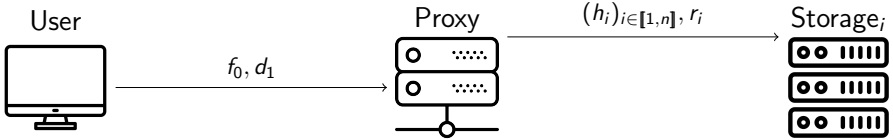
$\mathcal{MKZT}$

Upload



$\mathcal{MKZT}$

Upload



$d_0||d_1 \leftarrow \mathsf{AONT.Hide}(m)$

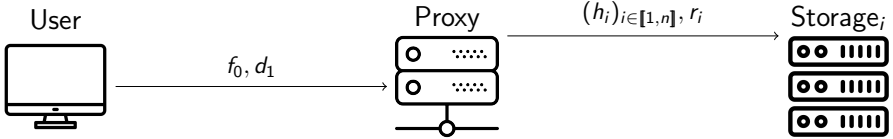
$f_0 \leftarrow \mathsf{MKE.Enc}((pk_i)_{i\in[\![1,n]\!]}, d_0)$

$h_1, \ldots, h_n \leftarrow \mathsf{SS.Split}(f_0)$

$r_1, \ldots, r_n \leftarrow \mathsf{SS.Split}(d_1)$

$\mathcal{MKZT}$

Upload



$d_0||d_1 \leftarrow \mathsf{AONT.Hide}(m)$

$f_0 \leftarrow \mathsf{MKE.Enc}((pk_i)_{i\in[\![1,n]\!]}, d_0)$

$h_1, \ldots, h_n \leftarrow \mathsf{SS.Split}(f_0)$

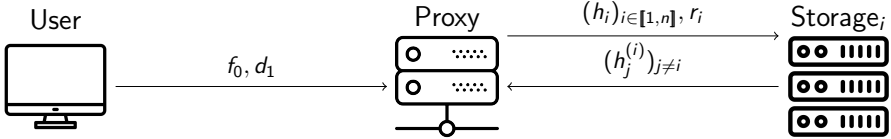
$r_1, \ldots, r_n \leftarrow \mathsf{SS.Split}(d_1)$

$\forall j \in [\![1, n]\!]$,

$h_j^{(i)} \leftarrow \mathsf{MKE.PartDec}(sk_i, h_j)$

$\mathcal{MKZT}$

Upload



$d_0||d_1 \leftarrow \mathsf{AONT.Hide}(m)$

$f_0 \leftarrow \mathsf{MKE.Enc}((pk_i)_{i\in[\![1,n]\!]}, d_0)$

$h_1, \ldots, h_n \leftarrow \mathsf{SS.Split}(f_0)$

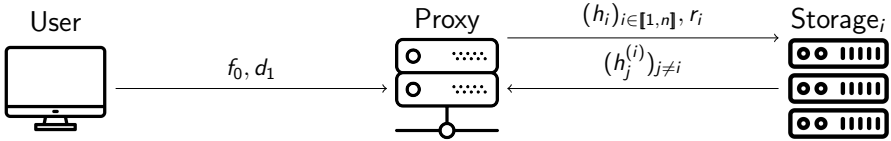
$r_1, \ldots, r_n \leftarrow \mathsf{SS.Split}(d_1)$

$\forall j \in [\![1, n]\!]$,

$h_j^{(i)} \leftarrow \mathsf{MKE.PartDec}(sk_i, h_j)$

$\mathcal{MKZT}$

Upload



$d_0||d_1 \leftarrow \mathsf{AONT.Hide}(m)$

$f_0 \leftarrow \mathsf{MKE.Enc}((pk_i)_{i\in[\![1,n]\!]}, d_0)$

$h_1, \ldots, h_n \leftarrow \mathsf{SS.Split}(f_0)$

$r_1, \ldots, r_n \leftarrow \mathsf{SS.Split}(d_1)$

$\forall j \in [\![1, n]\!]$,

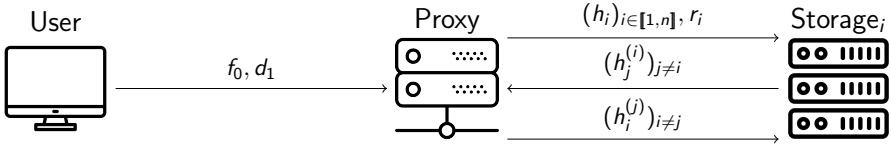
$h_j^{(i)} \leftarrow \mathsf{MKE.PartDec}(sk_i, h_j)$

$y_i \leftarrow \mathsf{MKE.FinDec}((h_i^{(j)})_{j\in[\![1,n]\!]})$

store r_i, y_i

$\mathcal{MKZT}$

Upload



$d_0||d_1 \leftarrow \mathsf{AONT.Hide}(m)$

$f_0 \leftarrow \mathsf{MKE.Enc}((pk_i)_{i\in[\![1,n]\!]}, d_0)$

$h_1, \ldots, h_n \leftarrow \mathsf{SS.Split}(f_0)$

$r_1, \ldots, r_n \leftarrow \mathsf{SS.Split}(d_1)$

$\forall j \in [\![1, n]\!]$,

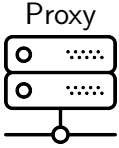
$h_j^{(i)} \leftarrow \mathsf{MKE.PartDec}(sk_i, h_j)$

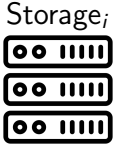
$y_i \leftarrow \mathsf{MKE.FinDec}((h_i^{(j)})_{j\in[\![1,n]\!]})$

store r_i, y_i

$\mathcal{MKZT}$

download



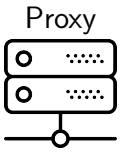


retrieve y_i, r_i

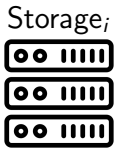
$h_i \leftarrow \mathsf{PKE.Enc}(pk_u, y_i)$

$\mathcal{MKZT}$

download



h_i, r_i

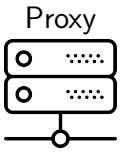


retrieve y_i, r_i

$h_i \leftarrow \mathrm{PKE.Enc}(pk_u, y_i)$

$\mathcal{MKZT}$

download



h_i, r_i

$f_0 \leftarrow \mathrm{SS.Rec}(h_1, \ldots, h_k)$

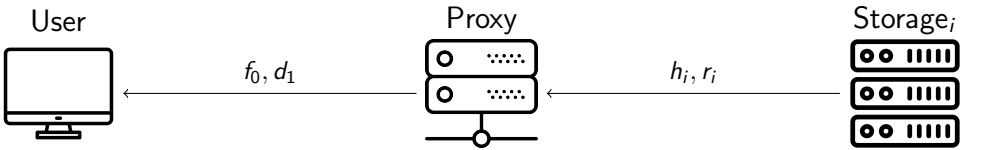
$d_1 \leftarrow \mathrm{SS.Rec}(r_1, \ldots, r_k)$

retrieve y_i, r_i

$h_i \leftarrow \mathrm{PKE.Enc}(pk_u, y_i)$

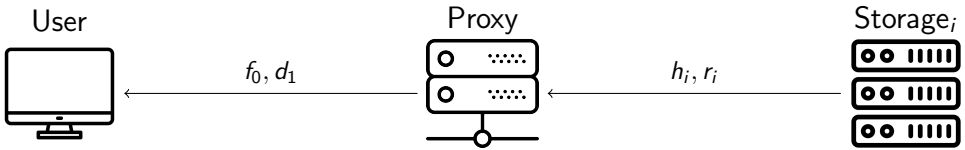
$\mathcal{MKZT}$

download

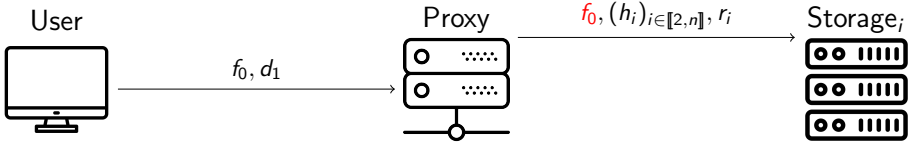


$\mathcal{MKZT}$

download



$\mathcal{MKZT}$is under attack



$d_0||d_1 \leftarrow \text{AONT.}\texttt{Hide}(m)$

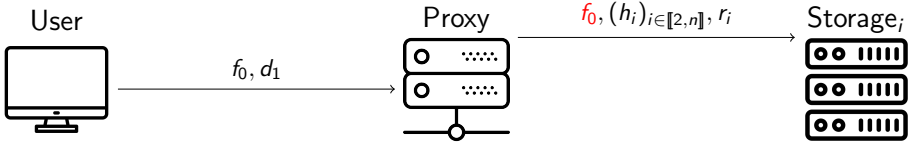
$f_0 \leftarrow \text{MKE.}\texttt{Enc}((pk_i)_{i\in[\![1,n]\!]}, d_0)$

$h_1, \ldots, h_n \leftarrow \text{SS.}\texttt{Split}(f_0)$

$r_1, \ldots, r_n \leftarrow \text{SS.}\texttt{Split}(d_1)$

$f_0^{(0)} \leftarrow \text{MKE.PartDec}(sk_0, f_0)$

$\mathcal{MKZT}$is under attack



$d_0||d_1 \leftarrow \text{AONT.Hide}(m)$

$f_0 \leftarrow \text{MKE.Enc}((pk_i)_{i\in[\![1,n]\!]}, d_0)$

$h_1, \ldots, h_n \leftarrow \text{SS.Split}(f_0)$

$r_1, \ldots, r_n \leftarrow \text{SS.Split}(d_1)$

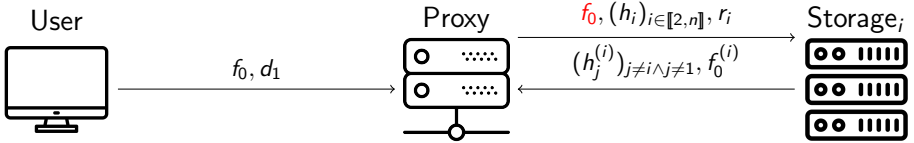
$f_0^{(0)} \leftarrow \text{MKE.PartDec}(sk_0, f_0)$

$\forall j \in [\![2, n]\!]$,

$h_j^{(i)} \leftarrow \text{MKE.PartDec}(sk_i, h_j)$

$f_0^{(i)} \leftarrow \text{MKE.PartDec}(sk_i, f_0)$

$\mathcal{MKZT}$is under attack



$d_0||d_1 \leftarrow \text{AONT.Hide}(m)$

$f_0 \leftarrow \text{MKE.Enc}((pk_i)_{i\in[\![1,n]\!]}, d_0)$

$h_1, \ldots, h_n \leftarrow \text{SS.Split}(f_0)$

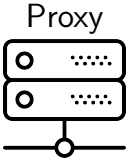
$r_1, \ldots, r_n \leftarrow \text{SS.Split}(d_1)$

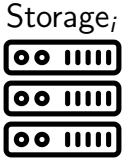
$f_0^{(0)} \leftarrow \text{MKE.PartDec}(sk_0, f_0)$

$d_0 \leftarrow \text{MKE.FinDec}((f_0^{(j)})_{j\in[\![1,n]\!]})$

$m \leftarrow \text{AONT.Rec}(d_0||d_1)$

$\mathcal{MKZT}$is under attack again

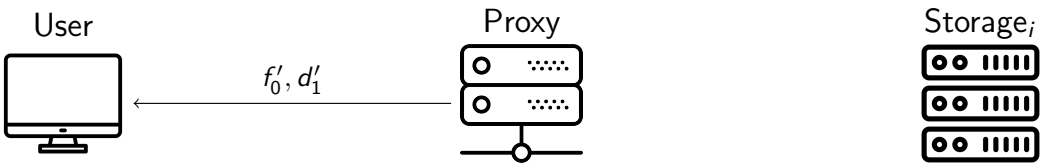




$d'_0, d'_1 \leftarrow \mathrm{AONT.Hide}(m')$

$f'_0 \leftarrow \mathrm{PKE.Enc}(pk_u, d'_0)$

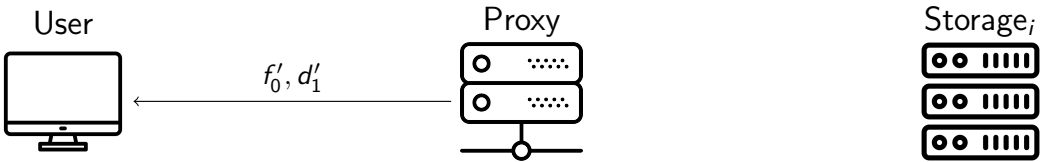
$\mathcal{MKZT}$is under attack again



$d'_0, d'_1 \leftarrow \text{AONT.Hide}(m')$

$f'_0 \leftarrow \text{PKE.Enc}(pk_u, d'_0)$

$\mathcal{MKZT}$is under attack again

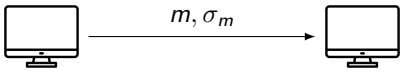


$\mathcal{PRZT}+$: (Improved) Proxy Re-Encryption-based Protocol

scheme	#Users	Proxy model	#Collusions	Secrecy	Integrity
$\mathcal{PRZT}$	1	Honest but Curious	0	✓	✓
$\mathcal{MKZT}$	1	Honest but Curious	k-1	✓	✓
$\mathcal{PRZT}+$	1	Malicious	k-1	✓	✓

Primitive	Introduced	Algorithms
Secret Sharing (SS)	✓	`Split`, `Merge`
Public Key Encryption (PKE)	✓	`KeyGen`, `Enc`, `Dec`
Signature (Sig)		`KeyGen`, `Sign`, `Verif`
AONT	✓	`Hide`, `Rec`

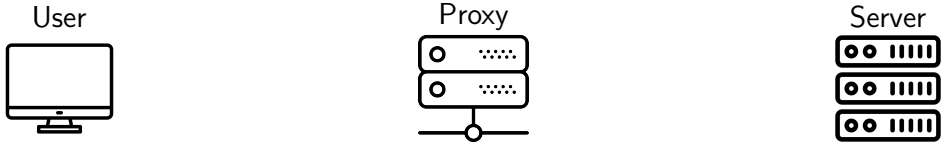
Signature



$\sigma_m \leftarrow \mathtt{Sign}(\mathsf{sk}, m)$ $\quad$ $\top/\bot \leftarrow \mathtt{Verif}(\mathsf{pk}, m, \sigma_m)$

$\mathcal{PRZT}+$

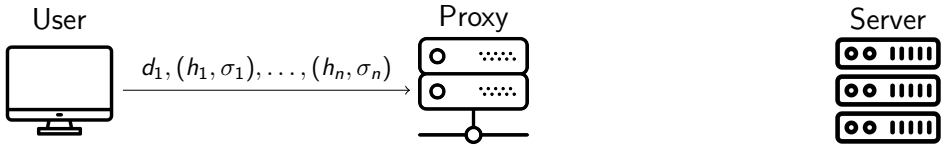
Upload



$$d_0||d_1 \leftarrow \text{AONT.}\texttt{Hide}(m)$$
$$s_1, \ldots, s_n \leftarrow \text{SS.}\texttt{Split}(d_0, n, k)$$
$$h_i \leftarrow \text{PKE.}\texttt{Enc}(pk_i, s_i)$$
$$\sigma_i \leftarrow \text{Sig.}\texttt{Sign}(sk_u, h_i)$$

$\mathcal{PRZT}+$

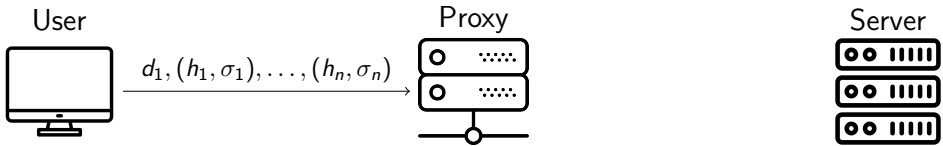
Upload



$$d_0 || d_1 \leftarrow \text{AONT.}\texttt{Hide}(m)$$
$$s_1, \ldots, s_n \leftarrow \text{SS.}\texttt{Split}(d_0, n, k)$$
$$h_i \leftarrow \text{PKE.}\texttt{Enc}(pk_i, s_i)$$
$$\sigma_i \leftarrow \text{Sig.}\texttt{Sign}(sk_u, h_i)$$

$\mathcal{PRZT}+$

Upload



$d_0 || d_1 \leftarrow \text{AONT.Hide}(m)$

$s_1, \ldots, s_n \leftarrow \text{SS.Split}(d_0, n, k)$

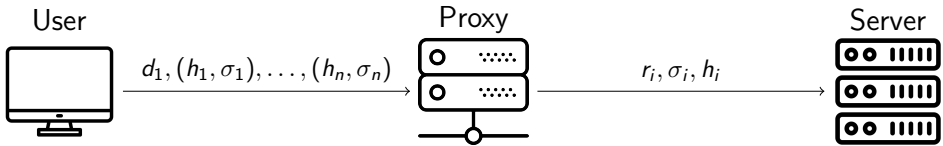
$h_i \leftarrow \text{PKE.Enc}(pk_i, s_i)$

$\sigma_i \leftarrow \text{Sig.Sign}(sk_u, h_i)$

$r_1, \ldots, r_n \leftarrow \text{SS.Split}(d_1)$

$\mathcal{PRZT}+$

Upload

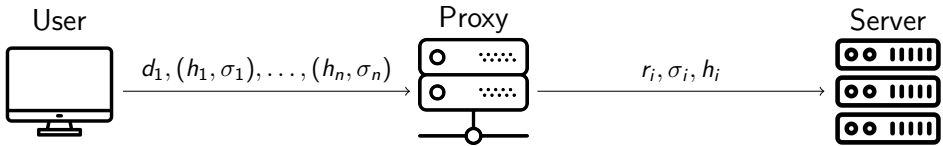


$$d_0 || d_1 \leftarrow \text{AONT.Hide}(m)$$
$$s_1, \ldots, s_n \leftarrow \text{SS.Split}(d_0, n, k)$$
$$h_i \leftarrow \text{PKE.Enc}(pk_i, s_i)$$
$$\sigma_i \leftarrow \text{Sig.Sign}(sk_u, h_i)$$

$$r_1, \ldots, r_n \leftarrow \text{SS.Split}(d_1)$$

$\mathcal{PRZT}+$

Upload



$d_0 || d_1 \leftarrow \text{AONT.Hide}(m)$

$s_1, \ldots, s_n \leftarrow \text{SS.Split}(d_0, n, k)$

$h_i \leftarrow \text{PKE.Enc}(pk_i, s_i)$

$\sigma_i \leftarrow \text{Sig.Sign}(sk_u, h_i)$

$r_1, \ldots, r_n \leftarrow \text{SS.Split}(d_1)$

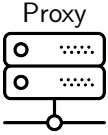
$\text{Sig.Verif}(pk_u, \sigma_i, h_i,)$

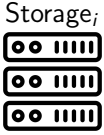
$y_i \leftarrow \text{PKE.Dec}(sk_i, h_i)$

store r_i, y_i

$\mathcal{PRZT}+$

Download



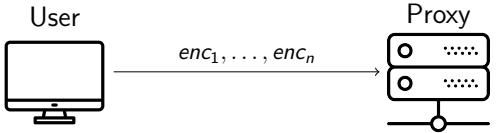


$nc_1, \ldots, nc_n \leftarrow \mathsf{PKE.Gen}(\lambda)$

$\forall i \in [\![1, n]\!], enc_i \leftarrow \mathsf{PKE.Enc}(pk_i, nc_i)$

$\mathcal{PRZT}+$

Download

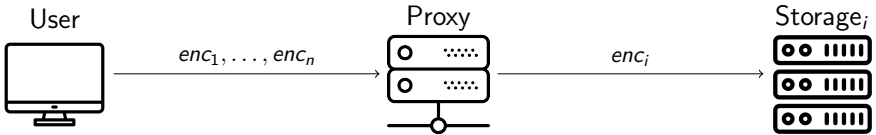


$nc_1, \ldots, nc_n \leftarrow \text{PKE.Gen}(\lambda)$

$\forall i \in [\![1, n]\!], enc_i \leftarrow \text{PKE.}\texttt{Enc}(pk_i, nc_i)$

$\mathcal{PRZT}+$

Download

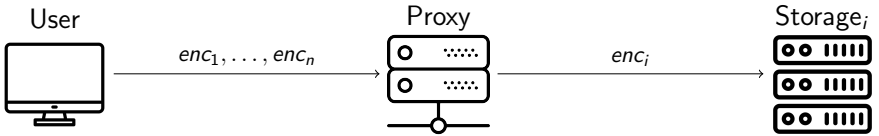


$nc_1, \ldots, nc_n \leftarrow \text{PKE.Gen}(\lambda)$

$\forall i \in [\![1, n]\!], enc_i \leftarrow \text{PKE.}\texttt{Enc}(pk_i, nc_i)$

$\mathcal{PRZT}+$

Download



$nc_1, \ldots, nc_n \leftarrow \text{PKE.Gen}(\lambda)$

$\forall i \in [\![1, n]\!], enc_i \leftarrow \text{PKE.Enc}(pk_i, nc_i)$

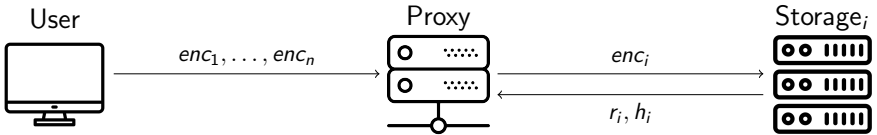
$nc_i \leftarrow \text{PKE.Dec}(sk_i, enc_i)$

retrieve y_i, r_i

$h_i \leftarrow \text{PKE.Enc}(pk_u, y_i || nc_i)$

$\mathcal{PRZT}+$

Download



$nc_1, \ldots, nc_n \leftarrow \text{PKE.Gen}(\lambda)$

$\forall i \in [\![1, n]\!], enc_i \leftarrow \text{PKE.Enc}(pk_i, nc_i)$

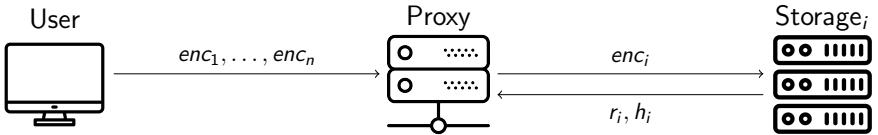
$nc_i \leftarrow \text{PKE.Dec}(sk_i, enc_i)$

retrieve y_i, r_i

$h_i \leftarrow \text{PKE.Enc}(pk_u, y_i || nc_i)$

$\mathcal{PRZT}+$

Download



$nc_1, \ldots, nc_n \leftarrow \text{PKE.Gen}(\lambda)$

$\forall i \in [\![1, n]\!], enc_i \leftarrow \text{PKE.Enc}(pk_i, nc_i)$

$d_1 \leftarrow \text{SS.Merge}(r_1, \ldots, r_n)$

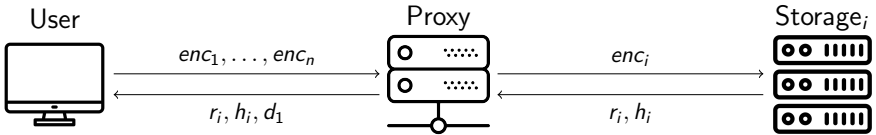
$nc_i \leftarrow \text{PKE.Dec}(sk_i, enc_i)$

retrieve y_i, r_i

$h_i \leftarrow \text{PKE.Enc}(pk_u, y_i || nc_i)$

$\mathcal{PRZT}+$

Download



$nc_1, \ldots, nc_n \leftarrow \text{PKE.Gen}(\lambda)$

$\forall i \in [\![1, n]\!], enc_i \leftarrow \text{PKE.Enc}(pk_i, nc_i)$

$d_1 \leftarrow \text{SS.Merge}(r_1, \ldots, r_n)$

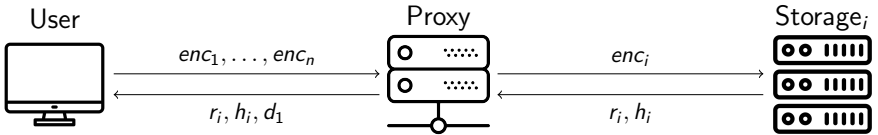
$nc_i \leftarrow \text{PKE.Dec}(sk_i, enc_i)$

retrieve y_i, r_i

$h_i \leftarrow \text{PKE.Enc}(pk_u, y_i || nc_i)$

$\mathcal{PRZT}+$

Download



$nc_1, \ldots, nc_n \leftarrow \text{PKE.Gen}(\lambda)$

$\forall i \in [\![1, n]\!], enc_i \leftarrow \text{PKE.Enc}(pk_i, nc_i)$

$d_1 \leftarrow \text{SS.Merge}(r_1, \ldots, r_n)$

$nc_i \leftarrow \text{PKE.Dec}(sk_i, enc_i)$

retrieve y_i, r_i

$h_i \leftarrow \text{PKE.Enc}(pk_u, y_i || nc_i)$

$\forall i \in [\![1, n]\!], y_i || nc_i \leftarrow \text{PKE.Dec}(sk_u, h_i)$

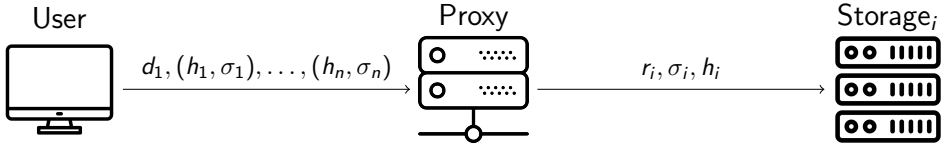
verify $(nc_i)_{i \in I}, I \subset [\![1, n]\!] \ \wedge |I| = k$

$d_0 \leftarrow \text{SS.Merge}(y_1, \ldots, y_k)$

$m \leftarrow \text{AONT.Rec}(d_0 || d_1)$

$\mathcal{PRZT}+$

Pros and cons



$d_0||d_1 \leftarrow \text{AONT.Hide}(m)$
$s_1, \ldots, s_n \leftarrow \text{SS.Split}(d_0, n, k)$
$h_i \leftarrow \text{PKE.Enc}(pk_i, s_i)$
$\sigma_i \leftarrow \text{Sig.Sign}(sk_u, h_i)$

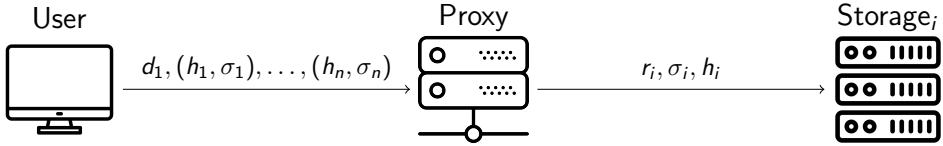
$r_1, \ldots, r_n \leftarrow \text{SS.Split}(d_1)$

$\text{Sig.Verif}(pk_u, \sigma_i, h_i,)$
$y_i \leftarrow \text{PKE.Dec}(sk_i, h_i)$
$store \quad r_i, y_i$

	#Users	Proxy model	#Collusions	Secrecy	Integrity	Authentication authority
$\mathcal{PRZT}+$	1	Malicious	k-1	✓	✓	✓

$\mathcal{PRZT}+$

Pros and cons



	#Users	Proxy model	#Collusions	Secrecy	Integrity	Authentication authority
$\mathcal{PRZT}+$	1	Malicious	k-1	✓	✓	✓

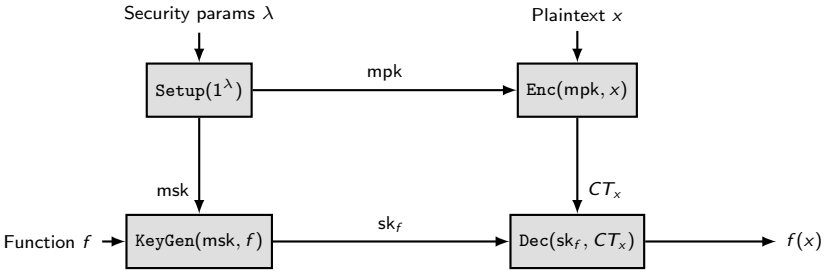
X: additional computation for the client.

$\mathcal{FEZT}$: Functional Encryption-based Protocol

scheme	#Users	Proxy model	#Collusions	Secrecy	Integrity
$\mathcal{PRZT}$	1	Honest but Curious	0	✓	✓
$\mathcal{MKZT}$	1	Honest but Curious	k-1	✓	✓
$\mathcal{PRZT}+$	1	Malicious	k-1	✓	✓
$\mathcal{FEZT}$	1	Malicious	k-1	✓	✓

Primitive	Introduced	Algorithms
Functional Encryption (FE)		`Setup`, `KeyGen`, `Enc`, `Dec`
Secret Sharing (SS)	✓	`Split`, `Merge`
Asymmetric Encryption (PKE)	✓	`KeyGen`, `Enc`, `Dec`
AONT	✓	`Hide`, `Rec`

Functional Encryption (FE)



Notions

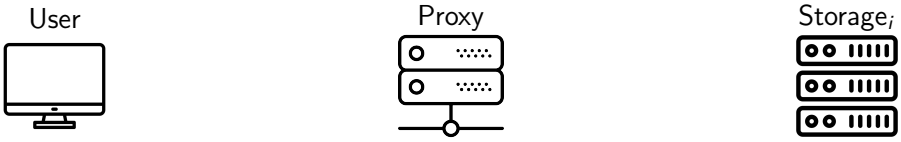
We denote the Functional Encryption scheme by $\{\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec}\}$

$$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathtt{Setup}(1^\lambda), \qquad CT_x \leftarrow \mathtt{Enc}(\mathsf{mpk}, x)$$

$$\mathsf{sk}_f \leftarrow \mathtt{KeyGen}(\mathsf{msk}, f), \qquad f(x) \leftarrow \mathtt{Dec}(\mathsf{sk}_f, CT_x)$$

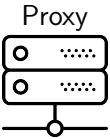
$\mathcal{FEZT}$

Upload



$\mathcal{FEZT}$

Upload



$d_0||d_1 \leftarrow \text{AONT.}\texttt{Hide}(m)$

$P(X), \xleftarrow{\$} \mathbb{Z}_p[X]^{k-1}$ st. $P(0) = 0$

$x_1, \ldots, x_n \xleftarrow{\$} \mathbb{Z}_p^*$ all distincts

$r_1, \ldots, r_n \xleftarrow{\$} \mathbb{Z}_p$

$\forall i \in [\![1, n]\!], \psi_{x_i} \leftarrow \text{PKE.}\texttt{Enc}(x_i||ri, pk_i)$

Let f a polynomial s.t

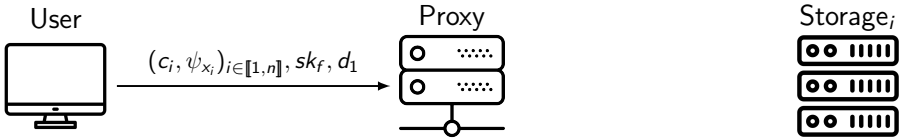
$$f(x_i||r_i||d_0) = P(x_i) + r_i + d_0$$

$sk_f \leftarrow \text{FE.KeyGen}(f, msk)$

$\forall i \in [\![1, n]\!], c_i \leftarrow \text{FE.}\texttt{Enc}(x_i||r_i||d_0, mpk)$

$\mathcal{FEZT}$

Upload



$d_0||d_1 \leftarrow \text{AONT.Hide}(m)$

$P(X), \xleftarrow{\$} \mathbb{Z}_p[X]^{k-1}$ st. $P(0) = 0$

$x_1, \ldots, x_n \xleftarrow{\$} \mathbb{Z}_p^*$ all distincts

$r_1, \ldots, r_n \xleftarrow{\$} \mathbb{Z}_p$

$\forall i \in [\![1, n]\!], \psi_{x_i} \leftarrow \text{PKE.Enc}(x_i||ri, pk_i)$

Let f a polynomial s.t

$$f(x_i||r_i||d_0) = P(x_i) + r_i + d_0$$

$sk_f \leftarrow \text{FE.KeyGen}(f, msk)$

$\forall i \in [\![1, n]\!], c_i \leftarrow \text{FE.Enc}(x_i||r_i||d_0, mpk)$

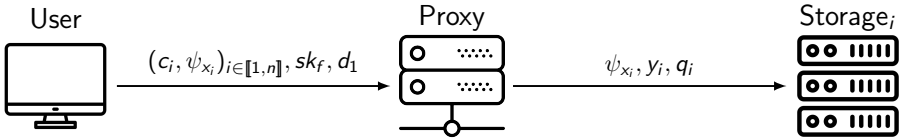
$\forall i \in [\![1, n]\!], y_i \leftarrow \text{FE.Dec}(c_i, sk_f)$

$(y_i = P(x_i) + r_i + d_0)$

$q_1, \ldots, q_n \leftarrow \text{SS.Split}(d_1, n, k)$

$\mathcal{FEZT}$

Upload



$d_0||d_1 \leftarrow \text{AONT.Hide}(m)$

$P(X), \xleftarrow{\$} \mathbb{Z}_p[X]^{k-1}$ st. $P(0) = 0$

$x_1, \ldots, x_n \xleftarrow{\$} \mathbb{Z}_p^*$ all distincts

$r_1, \ldots, r_n \xleftarrow{\$} \mathbb{Z}_p$

$\forall i \in [\![1, n]\!], \psi_{x_i} \leftarrow \text{PKE.Enc}(x_i||ri, pk_i)$

Let f a polynomial s.t

$f(x_i||r_i||d_0) = P(x_i) + r_i + d_0$

$sk_f \leftarrow \text{FE.KeyGen}(f, msk)$

$\forall i \in [\![1, n]\!], c_i \leftarrow \text{FE.Enc}(x_i||r_i||d_0, mpk)$

$\forall i \in [\![1, n]\!], y_i \leftarrow \text{FE.Dec}(c_i, sk_f)$

$(y_i = P(x_i) + r_i + d_0)$

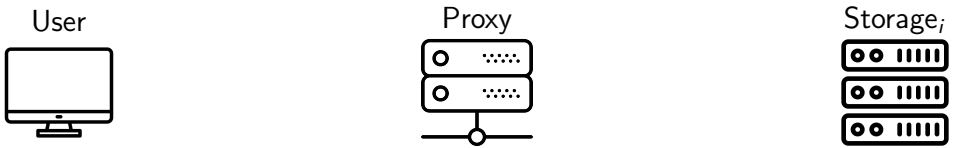
$q_1, \ldots, q_n \leftarrow \text{SS.Split}(d_1, n, k)$

$x_i||r_i \leftarrow \text{PKE.Dec}(\psi_{x_i}, sk_i)$

store $x_i, y_i - r_i, q_i$

$\mathcal{FEZT}$

Download

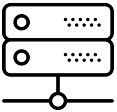


$\mathcal{FEZT}$

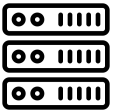
Download

User

Proxy



Storage$_i$



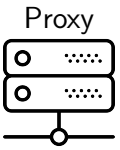
retrieve x_i, y_i, q_i

$\psi_{x_i} \leftarrow \mathsf{PKE.Enc}(x_i, pk_u)$

$\psi_{y_i} \leftarrow \mathsf{PKE.Enc}(y_i, pk_u)$

$\mathcal{FEZT}$

Download



$(\psi_{x_i}, \psi_{y_i}, q_i)$

$d_1 \leftarrow \mathsf{SS.Rec}(q_1, \ldots, q_n)$

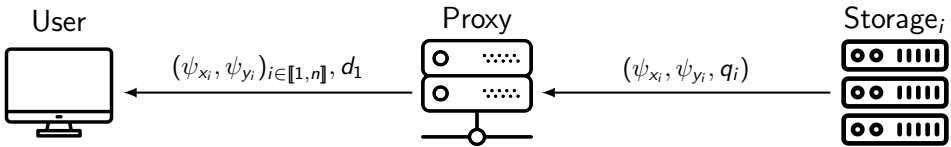
retrieve x_i, y_i, q_i

$\psi_{x_i} \leftarrow \mathsf{PKE.Enc}(x_i, pk_u)$

$\psi_{y_i} \leftarrow \mathsf{PKE.Enc}(y_i, pk_u)$

$\mathcal{FEZT}$

Download



$x_i \leftarrow \mathsf{PKE.Dec}(\psi_{x_i}, sk_u)_{i\in[\![1,n]\!]}$

$y_i \leftarrow \mathsf{PKE.Dec}(\psi_{y_i}, sk_u)_{i\in[\![1,n]\!]}$

$d_0 \leftarrow \mathsf{SS.Rec}((x_i, y_i)_{i\in[\![1,n]\!]})$

$m \leftarrow \mathsf{AONT.Rec}(d_0||d_1)$

$d_1 \leftarrow \mathsf{SS.Rec}(q_1, \ldots, q_n)$

retrieve x_i, y_i, q_i

$\psi_{x_i} \leftarrow \mathsf{PKE.Enc}(x_i, pk_u)$

$\psi_{y_i} \leftarrow \mathsf{PKE.Enc}(y_i, pk_u)$

Functional Encryption

Pros and cons

scheme	#Users	Proxy model	#Collusions	Secrecy	Integrity	Authentication authority
FE	1	Malicious	k-1	✓	✓	✓

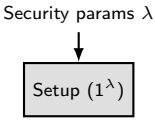
$\mathcal{PALAFOUR}$: Attribute-based Protocol

scheme	#Users	Proxy model	#Collusions	Secrecy	Integrity
$\mathcal{PRZT}$	1	Honest but Curious	0	✓	✓
$\mathcal{MKZT}$	1	Honest but Curious	k-1	✓	✓
$\mathcal{PRZT}+$	1	Malicious	k-1	✓	✓
$\mathcal{FEZT}$	1	Malicious	k-1	✓	✓
$\mathcal{PALAFOUR}$	>1	Malicious	n	✓	✓

Primitive	Introduced	Algorithms
Secret Sharing (SS)	✓	`Split, Merge`
Symmetric Encryption (Sym)	✓	`KeyGen, Enc, Dec`
Attribute-Based Encryption (ABE)		`Setup, KeyGen, Enc, Dec`
Identity-Based Signature (IBS)		`Setup, KeyGen, Sign, Verif`

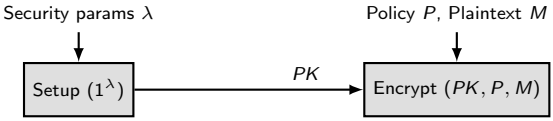
Attribute Based Encryption (ABE)

Ciphertext Policy ABE



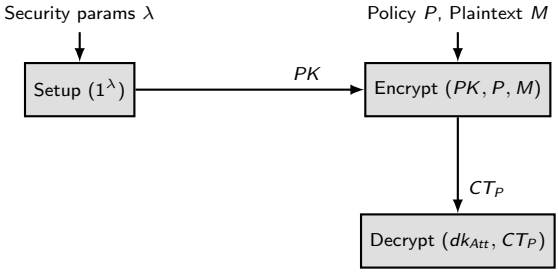
Attribute Based Encryption (ABE)

Ciphertext Policy ABE



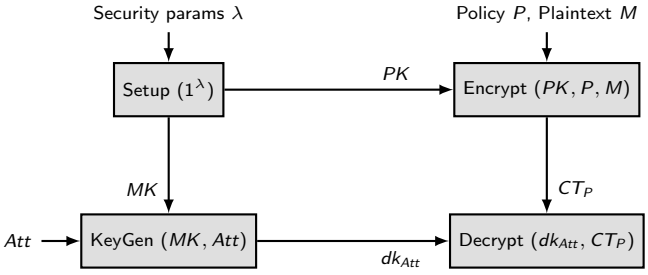
Attribute Based Encryption (ABE)

Ciphertext Policy ABE



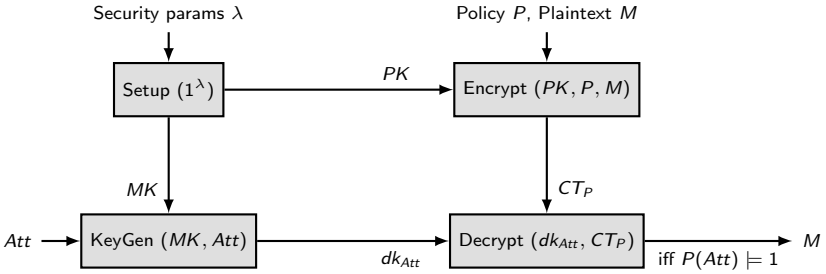
Attribute Based Encryption (ABE)

Ciphertext Policy ABE



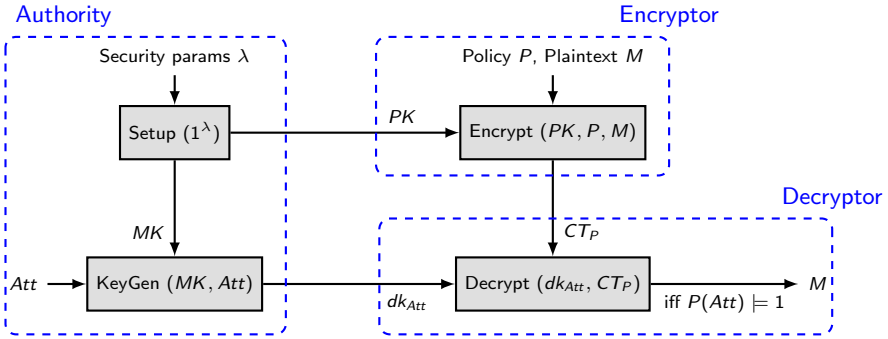
Attribute Based Encryption (ABE)

Ciphertext Policy ABE



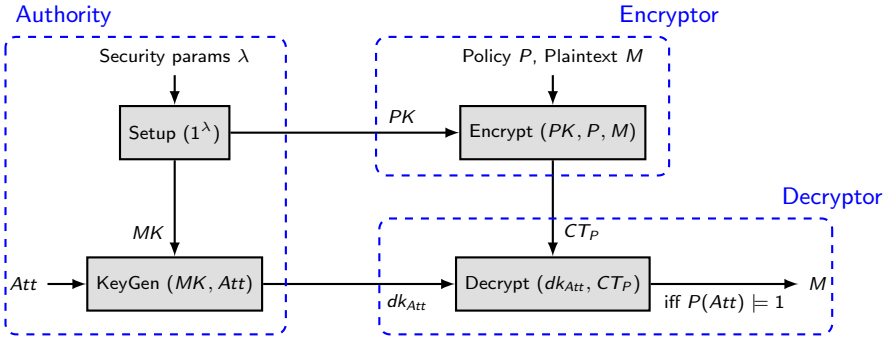
Attribute Based Encryption (ABE)

Ciphertext Policy ABE



Attribute Based Encryption (ABE)

Ciphertext Policy ABE



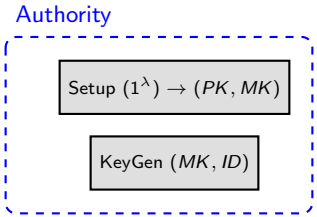
Notions

We denote the ABE scheme by $\mathsf{ABE} = \{\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec}\}$

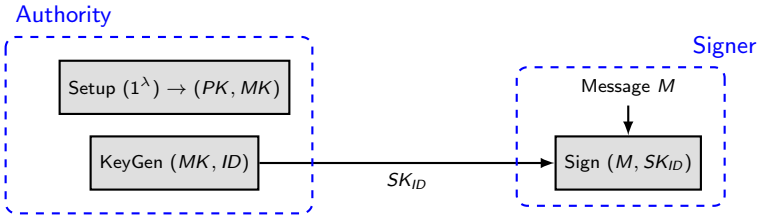
$$PK_{\mathsf{ABE}}, MK_{\mathsf{ABE}} \leftarrow \mathsf{ABE}.\mathtt{Setup}(1^\lambda), \qquad C_P \leftarrow \mathsf{ABE}.\mathtt{Enc}(PK_{\mathsf{ABE}}, P, M)$$

$$dk^{Att}_{\mathsf{ABE}} \leftarrow \mathsf{ABE}.\mathtt{KeyGen}(MK_{\mathsf{ABE}}, Att), \qquad \mathsf{ABE}.\mathtt{Dec}(dk^{Att}_{\mathsf{ABE}}, C_P)$$

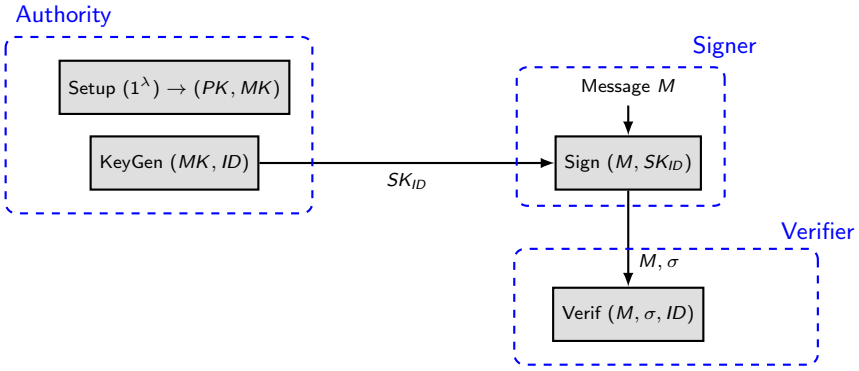
Identity Based Signature (IBS)



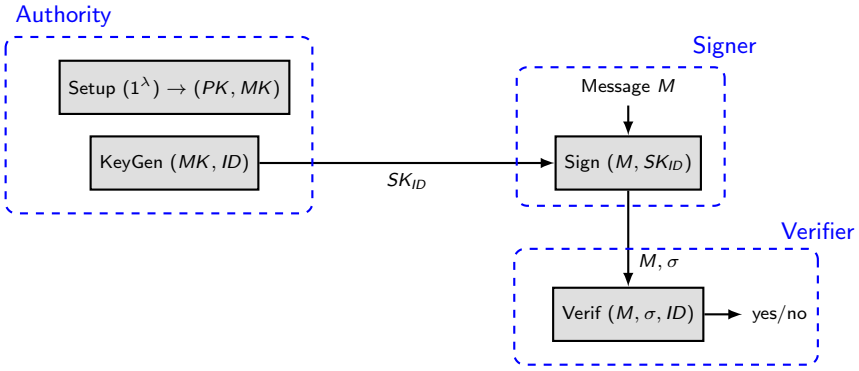
Identity Based Signature (IBS)



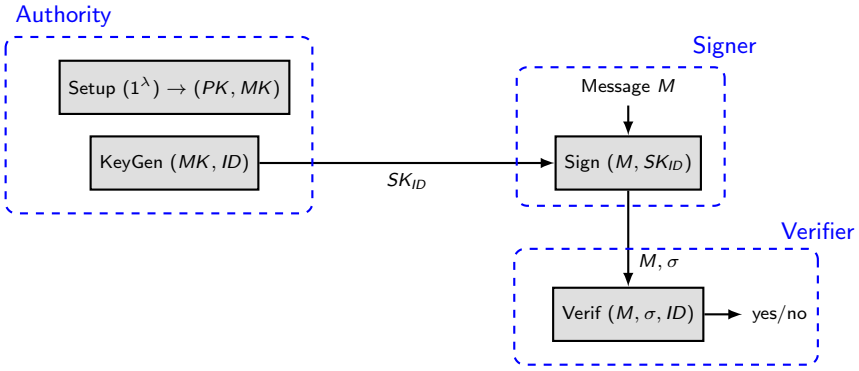
Identity Based Signature (IBS)



Identity Based Signature (IBS)



Identity Based Signature (IBS)



Notions

We denote the IBS scheme by IBS $= \{\texttt{Setup}, \texttt{KeyGen}, \texttt{Sign}, \texttt{Verif}\}$

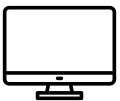
$$PK_{\text{IBS}}, MK_{\text{IBS}} \leftarrow \text{IBS}.\texttt{Setup}(1^\lambda), \qquad SK^{ID}_{\text{IBS}} \leftarrow \text{IBS}.\texttt{KeyGen}(MK_{\text{IBS}}, ID)$$

$$M, \sigma \leftarrow \text{IBS}.\texttt{Sign}(M, SK^{ID}_{\text{IBS}}), \qquad \text{IBS}.\texttt{Verif}(M, \sigma, ID)$$

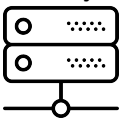
$\mathcal{PALAFOUR}$

Upload

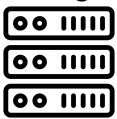
User



Proxy



Storage_i



$SK_{ID}^{\text{IBS}} \leftarrow \text{IBS.KeyGen}(MK, ID)$

$K \leftarrow Sym.\text{KeyGen}()$

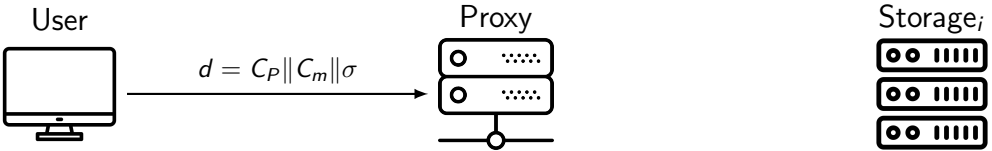
$C_p \leftarrow \text{ABE.Enc}(PK_{\text{ABE}}, P)$

$C_m \leftarrow \text{Sym.Enc}(m, K)$

$\sigma \leftarrow \text{IBS.Sign}(C_P \| C_m, SK_{ID}^{IBS})$

$PALAFOUR$

Upload



$SK_{ID}^{\text{IBS}} \leftarrow \text{IBS.KeyGen}(MK, ID)$

$K \leftarrow Sym.\text{KeyGen}()$

$C_p \leftarrow \text{ABE.Enc}(PK_{\text{ABE}}, P)$

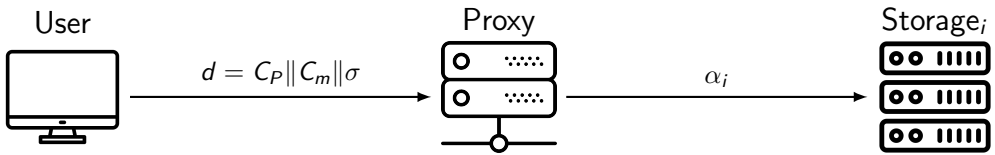
$C_m \leftarrow \text{Sym.Enc}(m, K)$

$\sigma \leftarrow \text{IBS.Sign}(C_P \| C_m, SK_{ID}^{IBS})$

$\alpha_1, \ldots, \alpha_n \leftarrow SS.Split(d, n, k)$

$PALAFOUR$

Upload



$SK_{ID}^{\text{IBS}} \leftarrow \text{IBS.KeyGen}(MK, ID)$

$K \leftarrow Sym.\text{KeyGen}()$

$C_p \leftarrow \text{ABE.Enc}(PK_{\text{ABE}}, P)$

$C_m \leftarrow \text{Sym.Enc}(m, K)$

$\sigma \leftarrow \text{IBS.Sign}(C_P \| C_m, SK_{ID}^{IBS})$

$\alpha_1, \ldots, \alpha_n \leftarrow SS.Split(d, n, k)$

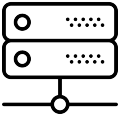
store α_i

$\mathcal{PALAFOUR}$

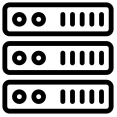
Download

User

Proxy



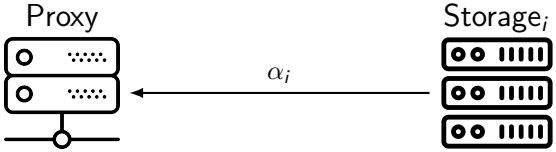
Storage$_i$



retrieve α_i

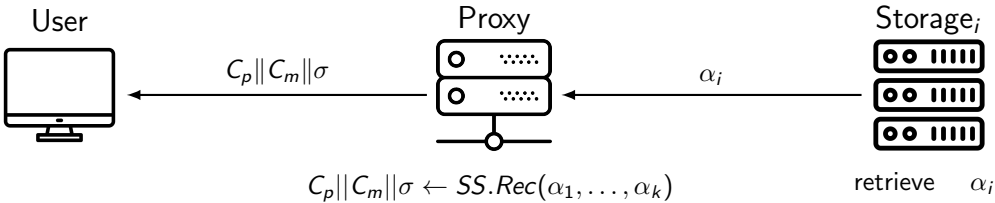
$\mathcal{PALAFOUR}$

Download



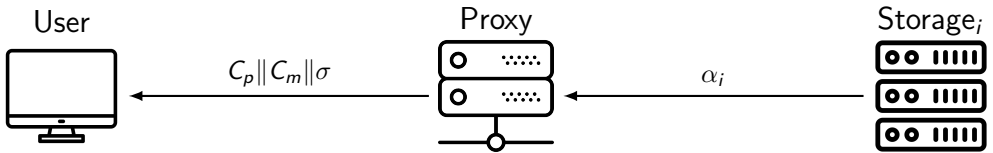
$PALAFOUR$

Download



$\mathcal{PALAFOUR}$

Download



Instances

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [Wat11]

- Elliptic curve, pairing-based
- Expressive
- Efficient
- Implemented by OpenABE [Zeu]

Instances

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [Wat11]

- Elliptic curve, pairing-based
- Expressive
- Efficient
- Implemented by OpenABE [Zeu]

Identity-Based Signature (IBS) [BB04]

- Elliptic curve, pairing-based
- Short signature
- Efficient ....
- Implemented by RELIC [AGM+]

Comparison

scheme	#Users	Proxy model	#Collusions	Secrecy	Integrity
$\mathcal{PRZT}$	1	Honest but Curious	0	✓	✓
$\mathcal{MKZT}$	1	Honest but Curious	k-1	✓	✓
$\mathcal{PRZT}+$	1	Malicious	k-1	✓	✓
$\mathcal{FEZT}$	1	Malicious	k-1	✓	✓
$\mathcal{PALAFOUR}$	>1	Malicious	n	✓	✓

Comparison

Scheme		Messages exchanges	
		Client↔Proxy	Proxy ↔ one Provider
$\mathcal{PRZT}$	Upload	$\|m\| + \|PRE\ cipher\|$	$\frac{\|m\|}{k} + \|PRE\ cipher\|$
	Download	$\|m\| + \|PRE\ cipher\|$	$\frac{\|m\|}{k} + \|PRE\ cipher\|$
$\mathcal{MKZT}$	Upload	$\|m\| + \|MKE\ cipher\|$	$\frac{\|m\|}{k} + 3 * (n-1) * \|MKE\ cipher\|$
	Download	$\|m\| + \|MKE\ cipher\|$	$\frac{\|m\|}{k} + \|MKE\ cipher\|$
$\mathcal{PRZT}+$	Upload	$\|m\| + n * \|Asym\ cipher\| + n * \|Asymsignature\|$	$\frac{\|m\|}{k} + \|Asym\ cipher\| + \|Asymsignature\|$
	Download	$\|m\| + n * \|nonce\ Asym\ cipher\| + n * \|Asym\ cipher\|$	$\frac{\|m\|}{k} + \|nonce\ Asym\ cipher\| + \|Asym\ cipher\|$
$\mathcal{FEZT}$	Upload	$\|m\| + n * \|FE\ cipher\| + n * \|Asym\ Cipher\| + \|FE\ function\ key\|$	$\frac{\|m\|}{k} + \|Asym\ Cipher\| + \|C\|$
	Download	$\|m\| + n * \|FE\ cipher\| + 2 * \|Asym\ Cipher\|$	$\frac{\|m\|}{k} + 2 * \|Asym\ Cipher\|$
$\mathcal{PALAFOUR}$	Upload Download	$\|\text{signature IBS}\| + \|C * (policy\ size)\| + \|m\| = d$	$\frac{d}{k}$

Conclusion

Contributions

- 1 Attack on $\mathcal{PRZT}$

Conclusion

Contributions

- 1 Attack on $\mathcal{PRZT}$
- 2 Attacks on $\mathcal{MKZT}$

Conclusion

Contributions

- 1 Attack on $\mathcal{PRZT}$
- 2 Attacks on $\mathcal{MKZT}$
- $\mathcal{PRZT}+$

Conclusion

Contributions

- 1 Attack on $\mathcal{PRZT}$
- 2 Attacks on $\mathcal{MKZT}$
- $\mathcal{PRZT}+$
- $\mathcal{FEZT}$

Conclusion

Contributions

- 1 Attack on $\mathcal{PRZT}$
- 2 Attacks on $\mathcal{MKZT}$
- $\mathcal{PRZT}+$
- $\mathcal{FEZT}$
- $\mathcal{PALAFOUR}$

Conclusion

Contributions

- 1 Attack on $\mathcal{PRZT}$
- 2 Attacks on $\mathcal{MKZT}$
- $\mathcal{PRZT}+$
- $\mathcal{FEZT}$
- $\mathcal{PALAFOUR}$

Future Work

- Post-Quantum

Conclusion

Contributions

- 1 Attack on $\mathcal{PRZT}$
- 2 Attacks on $\mathcal{MKZT}$
- $\mathcal{PRZT}+$
- $\mathcal{FEZT}$
- $\mathcal{PALAFOUR}$

Future Work

- Post-Quantum
- Authority removal

References

[BB04] Dan Boneh et Xavier Boyen :
Short signatures without random oracles.
In International conference on the theory and applications of cryptographic techniques, pages 56–73. Springer, 2004.

[Wat11] Brent Waters :
Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization.
In International workshop on public key cryptography, pages 53–70. Springer, 2011.