D4.3 Annual Report on Functional Encryption Schemes for Prototypes Y3

Document Identification			
Status	Final	**Due Date**	31/03/2020
Version	1.0	**Submission Date**	31/03/2020

Related WP	WP4	**Document Reference**	D4.3
Related Deliverable(s)	D3.1, D4.1, D4.2, D7.1	**Dissemination Level(*)**	CO
Lead Participant	FUAS	**Lead Author**	Clément Gentilucci
Contributors	ENS, FUAS	**Reviewers**	Ward Beullens (KUL) Norman Scaife (WALLIX)

Keywords:
Functional Encryption Schemes, Digital Currency, Web Analytics, Video Surveillance

This document is issued within the frame and for the purpose of the FENTEC project. This project has received funding from the European Union's Horizon2020 under Grant Agreement No. 780108. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

(*) Dissemination level.-PU: Public, fully open, e.g. web; CO: Confidential, restricted under conditions set out in Model Grant Agreement; CI: Classified, Int = Internal Working Document, information as referred to in Commission Decision 2001/844/EC.

Document Information

List of Contributors	
Name	Partner
Michel Abdalla	ENS
Clément Gentilucci	FUAS

Document History			
Version	Date	Change editors	Changes
0.1	15/02/2020	Clément Gentilucci (FUAS)	ToC
0.2	28/02/2020	Clément Gentilucci (FUAS)	Added elements from D4.1 and D4.2
0.3	10/03/2020	Clément Gentilucci (FUAS)	First draft of Introduction and Conclusion
0.4	17/03/2020	Clément Gentilucci (FUAS), Michel Abdalla (ENS)	First complete version
0.5	17/03/2020	Clément Gentilucci (FUAS), Michel Abdalla (ENS)	Version for internal review
0.6	25/03/2020	Michel Abdalla (ENS)	Addressed review comments by WALLIX
0.7	30/03/2020	Michel Abdalla (ENS)	Addressed review comments by KUL
1.0	30/03/2020	Michel Abdalla (ENS)	Final version

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable Leader	Clément Gentilucci (FUAS)	31/03/2020
Technical Manager	Michel Abdalla (ENS)	31/03/2020
Quality Manager	Diego Esteban (ATOS)	31/03/2020
Project Coordinator	Francisco Gala (ATOS)	31/03/2020

Table of Contents

List of Figures

List of Acronyms

Acronym	Description
3-PDDH	3-party Decision Diffie-Hellman
ABE	Attribute-Based Encryption
DDH	Decisional Diffie-Hellman
DMCFE	Decentralized Multi-Client Functional Encryption
FAME	Fast Attribute-based Message Encryption
FE	Functional Encryption
IND-CPA	Indistinguishability under Chosen-Plaintext Attacks
IPPE	Inner Product Predicate Encryption
LDM	Local Decision Making
LWE	Learning With Errors
MCFE	Multi-Client Functional Encryption
MDDH	Matrix Decisional Diffie-Hellman
PE	Predicate Encryption
PPT	Probabilistic Polynomial Time
ROM	Random-Oracle Model
SE	Symmetric Encryption

Executive Summary

This deliverable provides updated and final specifications of the practical functional encryption schemes tailored to the different use cases. It also serves as a basis to WP5 as a key element to determine fitting hardware solutions, to WP6 as primitives to be implemented in a software solution, and to WP7 as a building block for the first prototypes. To achieve this goal, this deliverable proposes cryptographic primitives for the three use cases in WP7 together with possible instantiations of these primitives.

Overall, there is no major change since D4.10 and the introduction of a labeled MCFE scheme [1] developed in the context of the FENTEC project. We believe the solutions presented are still the best available. Thus the deliverable will serve two purposes, reaffirming the motivation behind our choices and compiling the functional encryption solution we propose for each use case.

Section 4, which deals with the Digital Currency use case, describes the attribute-based encryption (ABE) scheme FAME [6], which had already been chosen in D4.1. FAME still represents the state of the art in terms of ABE schemes and is able to support the set of attributes needed by the Digital Currency use case.

Section 5 proposes two different solutions for this use case, depending on whether labels are needed. Regarding labeled schemes, two DMCFE schemes based on DDH and lattices are presented. The lattice-based scheme was developed in the context of the FENTEC project and was first presented in the D4.10, but not in previous versions of this deliverable (i.e., D4.1 and D4.2). This scheme is quantum-safe and more versatile but is less efficient than the DDH-based scheme. Regarding constructions without labels, we describe a scheme with strong security properties, such as adaptive security. It can be instantiated under different assumptions, such as plain DDH in pairing-free groups or lattice assumptions.

Section 6 re-introduces a DDH-based inner-product encryption scheme, but this time as our main solution. Indeed, recent development in this use case showed that the main limitation of DDH-based schemes, namely the ability to only decrypt small results, will not be a problem as we plan to limit ourselves to very few motion vectors in order to achieve motion detection.

Finally, Section 7 discusses the relations to requirements in D3.1 and the solutions proposed in D4.1 and D4.2.

1 Introduction

The FENTEC project works with an iterative life cycle. In this deliverable, we present the final choices of cryptographic primitives and instantiations that are to be used in the FENTEC API. We have updated the proposed solutions when necessary, by taking into account the evolving needs of each use case as well as progress in the field of functional encryption. When doing so, this deliverable supposes familiarity with the cryptographic primitives notions and solutions presented in D4.1 and D4.2, as well as the first and second annual reports on functional encryption schemes.

D4.1 and D4.2 described cryptographic solutions for the 3 uses cases in WP7: (1) Digital Currencies, (2) Web Analytics, and (3) Video Surveillance. We now summarize the changes with respect to these use cases:

Digital Currencies: Better understanding of the policies and the continuous survey of state-of-the-art functional encryption solutions make us believe that the two variants of the ABE scheme called FAME from [6] are still the best solutions;

Web Analytics: Academic research in the context of the FENTEC project brought up a new lattice-based DMCFE construction that supports encryption labels and whose security is based on a quantum-safe assumption; and

Video Surveillance: As already mentioned in D4.1, the solution based on exact threshold encryption is not optimal, since it assumes that the sequence of images within a video are encrypted bit by bit. In D4.2, we proposed two new solutions based on functional encryption schemes for the inner-product and quadratic-polynomial functionalities from [7, 8], which are more efficient and better match the requirements of the video encoding. This time, we only describe the scheme from [7] as some preprocessing seems mandatory.

1.1 Purpose of the Document

A primary goal of this deliverable is to pursue the investigation, started with the D4.1 and D4.2, of existing functional encryption schemes that meet the requirements of our use cases and to develop new schemes if necessary. Thus, this deliverable updates D4.2 by presenting new cryptographic primitives that improve upon multiple fundamental aspects, such as security, performance and practicality. The second goal of this deliverable is to be a compilation of the best current solutions we found until now for each use case. These schemes are to be implemented in Task 6.1 for the crypto API. Lastly, a final goal of this deliverable is to continue the survey of the literature in functional encryption to ensure that, by the end of the project, the cryptographic primitives being implemented are still the most relevant for our use cases.

1.2 Structure and Methodology

This document is strictly a summary of our choice for each use case. Thus, most of the definitions, basic tools, cryptographic primitives and security notions will be recalled. Regarding the cryptographic primitives for the use cases, we try to present only the best solutions. In this case, we will follow the same structure as in D4.1 and D4.2 by presenting the new cryptographic solution together with its security statement. Sections 4 to 6 are devoted to an in-depth exploration of the instantiations of the cryptographic schemes for all three use cases: the digital currency, the Web Analytics and Video Surveillance use cases. Multiple instantiations of cryptographic primitives are provided in these sections. The presentation of an instantiation starts with an explanation of the motivation behind the change that has been made. The instantiation itself is then presented together with a security statement for it. Section 7 then summarizes the list of use-case requirements that are met by the schemes described in Sections 4 to 6. Finally, Section 8 summarizes the main outcomes of this deliverable and discusses future research directions where improvements are needed.

2 Basic Tools

In this section, we recall some of the definitions and basic tools that will be used in the remainder of the document.

2.1 Notation and General Definitions

Let $\mathbb{N}$ denote the set of natural numbers. If $n \in \mathbb{N}$, then $\{0,1\}^n$ denotes the set of n-bit strings, and $\{0,1\}^*$ is the set of all bit strings. More generally, if S is a set, then S^n is the set of n-tuples of elements in S. If S is a set then $|S|$ denotes its size. If S is finite, then $x \leftarrow_{\$} S$ denotes the assignment to x of an element chosen uniformly at random from S. If $\mathcal{A}$ is an algorithm, then $y \leftarrow \mathcal{A}(x)$ denotes the assignment to y of the output of $\mathcal{A}$ on input x, and if $\mathcal{A}$ is randomized, then $y \leftarrow_{\$} \mathcal{A}(x)$ denotes that the output of an execution of $\mathcal{A}(x)$ with fresh randomness. "PPT" stands for probabilistic polynomial time. Most of the time we denote by λ the security parameter. A function $\nu : \mathbb{N} \to [0,1]$ is said to be *negligible* if for every $c \in \mathbb{N}$ there exists a $\lambda_c \in \mathbb{N}$ such that $\nu(\lambda) \leq \lambda^{-c}$ for all $\lambda > \lambda_c$.

In Section 5 and Section 6, we use implicit representation of group elements as introduced in [14]. That is, if $\mathbb{G}_1$ is a group of order p and g_1 a generator, then $\forall a \in \mathbb{Z}_p$, we note $[a]_1 = g_1^a$. If $A \in \mathbb{Z}_p^{m \times n}$ is a matrix, then $[A]_1 = (g_1^{a_{i,j}})_{1 \leq i \leq m, 1 \leq j \leq n}$.

The Matrix Decisional Diffie-Hellman (MDDH) Assumption [14], which we described later in this section, requires the following definition.

Definition 1 (Matrix Distribution) *Let $k \in \mathbb{N}$. We call D_k a matrix distribution if it outputs in polynomial time matrices in $\mathbb{Z}_p^{(k+1)\times k}$ of full rank k, and satisfying the following property,*

$$\Pr[\mathsf{orth}(\mathbf{A}) \subseteq \mathsf{span}(\mathbf{B})] = \frac{1}{\Omega(p)},$$

where $\mathbf{A}, \mathbf{B} \overset{R}{\leftarrow} \mathsf{D}_k$.

2.2 Represented groups

Let $\mathbb{G} = \langle g \rangle$ be a finite cyclic group of prime order p generated by an element g, where $\lambda = |p|$ is the security parameter. Throughout this report, we will use the multiplicative notation for the group operation. Hence, g^0 denotes the identity element of $\mathbb{G}$ and g^u denotes the group element of $\mathbb{G}$ that results from multiplying u copies of g, for $u \in \mathbb{N}$. Note that $g^u = g^{u \bmod |\mathbb{G}|}$ by Lagrange's theorem.

Algorithms which operate on $\mathbb{G}$ will be given string representations of elements in $\mathbb{G}$. For that, we require an injective map $\_ : \mathbb{G} \to \{0,1\}^\ell$ associated to $\mathbb{G}$, where ℓ is the length of the representation of group elements. Similarly, when a number $i \in \mathbb{N}$ is an input to, or output of, an algorithm, it must be appropriately encoded, say in binary. We assume all necessary encoding methods are fixed, and we do not normally write the _ operators.

The schemes considered in this report are parameterized by a *group generator*, which is a PTA *GroupGen* that on input 1^λ returns the description of a multiplicative group $\mathbb{G}$ of prime order p, where $2^\lambda < p < 2^{\lambda+1}$.

The choice of the security parameter will determine the exact security of schemes implemented over these groups. For instance, according to NIST's equivalence table [21], 160-bit discrete log subgroups and elliptic curve groups correspond to 80-bit symmetric keys and 1024-bit RSA/DLOG keys. Likewise, 256-bit discrete log subgroups and elliptic curve groups would correspond to 128-bit symmetric keys and 3072-bit RSA/DLOG keys. For other recommendations, please refer to the NIST's report

2.3 Bilinear maps

For a detailed introduction to pairings, see e.g. [11, Ch. IX].

The pairing-based schemes that we consider in this report are parameterized by a *pairing parameter generator*, which is a PTA *GroupGen* that on input 1^λ returns the description of three multiplicative groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ of order p for a 2λ-bit prime p together with generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ and an admissible map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$. By admissible, we mean that the map is bilinear, non-degenerate, and efficiently computable. Bi-linearity means that for all $a, b \in \mathbb{Z}_p^*$, we have $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$. By non-degenerate, we mean that $\hat{e}(g_1, g_2) \neq 1$.

In some cases, an efficient isomorphism is available from $\mathbb{G}_1$ into $\mathbb{G}_2$. This gives a symmetric pairing and we can use the notation $\mathbb{G} = \mathbb{G}_1 = \mathbb{G}_2$ to implicitly denote the use of the isomorphism in the pairing computation.

Finding optimal pairing-friendly elliptic curves is an active field of research (see [15]). At a 128-bit security level, the optimal choice would be to construct an elliptic curve whose order is a prime of 256 bits and over a prime finite field of the same size. Such optimal pairing-friendly curves exist [9] (Barreto-Naehrig (BN) curves), but usually have a special form [19].

2.4 Complexity Assumptions

We recall the definitions of some standard and non-standard complexity assumptions needed by the cryptographic schemes described in this report. For a more detailed description of the complexity assumptions used in cryptography, please refer to the final report on hard problems in cryptography from the ECRYPT2 project [22].

Definition 2 (Decisional Diffie-Hellman (DDH) assumption) *We say that the Decisional Diffie-Hellman assumption holds with respect to GroupGen*(1^λ) *if for all PPT adversaries* $\mathcal{A}$*, the following advantage function is negligible in* λ*:*

$$\mathsf{Adv}_{\mathcal{A},DDH}(\lambda) := |\Pr[\mathcal{A}(g^a, g^r, g^s) = 1] - \Pr[\mathcal{A}(g^a, g^r, g^{ar}) = 1]| < \epsilon(\lambda)$$

where $\mathbb{G} \leftarrow_\$ GroupGen(1^\lambda)$ *with generator* g*,* ϵ *is a negligible function, and* $a, r, s \leftarrow_\$ \mathbb{Z}_p$

Definition 3 (Learning-With-Errors (LWE) assumption) *Let* q, α, m *be functions of a parameter* n*. For a secret* $s \in \mathbb{Z}_q^n$*, the distribution* $A_{q,\alpha,s}$ *over* $\mathbb{Z}_q^n \times \mathbb{Z}_q$ *is obtained by sampling* $a \leftarrow_\$ \mathbb{Z}_q^n$ *and an error* $e \leftarrow_\$ \psi_{\mathbb{Z},\alpha,q}$ *from an error distribution* $\psi_{\mathbb{Z},\alpha,q}$*, and returning* $(a, \langle a, s\rangle + e) \in \mathbb{Z}_q^{n+1}$*. Let* $U(\mathbb{Z}_q^{m\times(n+1)})$ *denote the uniform distribution over* $\mathbb{Z}_q^{m\times(n+1)}$*. The Learning-With-Errors problem* $\mathsf{LWE}_{q,\alpha,m}$ *is as follows: For* $s \leftarrow_\$ \mathbb{Z}_q^n$*, the goal is to distinguish between the distributions:*

$$\mathsf{D}_0(s) := U(\mathbb{Z}_q^{m\times(n+1)}) \text{ and } \mathsf{D}_1(s) := (A_{q,\alpha,s})^m.$$

We say that a PPT algorithm $\mathcal{A}$ *solves the* $\mathsf{LWE}_{q,\alpha,m}$ *problem if it distinguishes* $\mathsf{D}_0(s)$ *and* $\mathsf{D}_1(s)$ *(with non-negligible advantage over the random coins of* $\mathcal{A}$ *and the randomness of the samples) with non-negligible probability over the randomness of* s*. The LWE assumption states that no such adversary exists.*

Definition 4 (D_k**-Matrix Diffie-Hellman Assumption** D_k-MDDH) *Let* D_k *be a matrix distribution. The* D_k*-Matrix Diffie-Hellman (*D_k-MDDH*) Assumption holds relative to* $\mathcal{G}$ *in* $\mathbb{G}_s$*, for* $s \in \{1, 2, T\}$*, if for all PPT adversaries* $\mathcal{A}$*,*

$$\mathsf{Adv}_{\mathcal{G},\mathcal{A}}^{\mathsf{D}_k\text{-MDDH}}(\lambda) := |\Pr[\mathcal{A}(\mathsf{bgp}, [\mathbf{A}]_s, [\mathbf{Aw}]_s) = 1] - \Pr[\mathcal{A}(\mathsf{bgp}, [\mathbf{A}]_s, [\mathbf{u}]_s) = 1]| = \mathsf{negl}(\lambda),$$

where probabilities are over $\mathsf{bgp} \xleftarrow{R} \mathcal{G}(1^\lambda)$, $\mathbf{A} \xleftarrow{R} \mathsf{D}_k, \mathbf{w} \xleftarrow{R} \mathbb{Z}_p^k, \mathbf{u} \xleftarrow{R} \mathbb{Z}_p^{k+1}$.

Definition 5 (3-party Decision Diffie-Hellman Assumption 3-PDDH**)** *We say that the 3-party Decision Diffie-Hellman Assumption (*3-PDDH*) Assumption holds relative to* $\mathcal{G}$ *if for all PPT adversaries* $\mathcal{A}$*,*

$$\begin{aligned}\mathsf{Adv}^{3\text{-PDDH}}_{\mathcal{G},\mathcal{A}}(\lambda) := |\,\Pr[\mathcal{A}(\mathsf{bgp}, [a]_1, [b]_2, [c]_1, [c]_2, [abc]_1) = 1] \\ - \Pr[\mathcal{A}(\mathsf{bgp}, [a]_1, [b]_2, [c]_1, [c]_2, [d]_1) = 1]| = \mathsf{negl}(\lambda)\end{aligned}$$

where the probability is taken over $\mathsf{bgp} \xleftarrow{R} \mathcal{G}(1^\lambda)$*,* $a, b, c, d \xleftarrow{R} \mathbb{Z}_p$.

3 Cryptographic Primitives

In this section, we recall the definitions of the cryptographic primitives used in the remainder of the document. Every primitive contains the definition for such a scheme as well as the security definitions associated with them.

3.1 Ciphertext-Policy Attribute-Based Encryption

Attribute-Based Encryption (ABE) [23, 18] belongs to the field of public-key cryptography. In an ABE scheme the secret key and the ciphertext are dependent upon attributes (e.g. a country, a role) and access policies. In such a system, the decryption of a ciphertext is possible, only if the set of attributes matches the access policy. These policies can be defined over the ciphertext, as well as over the key. Here characterize CP-ABE schemes according to this property:

Definition 6 (CP-ABE Scheme [23, 18]) *A CP-ABE scheme over a message space* $\mathcal{M}$ *is a tuple of four PPT algorithms* $\mathsf{CP\text{-}ABE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$*, such that*

- $\mathsf{Setup}(1^\lambda) \longrightarrow (\mathsf{pk}, \mathsf{msk})$*:* Setup *takes as input the security parameter* λ*. The algorithm outputs the public key* pk *and a master key* msk*.*
- $\mathsf{Enc}(\mathsf{pk}, \mathbb{A}, m) \longrightarrow c$*:* Enc *takes as inputs the public key* pk*, an access structure* $\mathbb{A}$ *and a message* $m \in \mathcal{M}$*. The algorithm outputs a ciphertext* c*.*
- $\mathsf{KeyGen}(\mathsf{msk}, S) \longrightarrow \mathsf{sk}$*:* KeyGen *takes as input the master key* msk *and a set of attributes* S*. The algorithm outputs a secret key* sk*.*
- $\mathsf{Dec}(\mathsf{pk}, c, \mathsf{sk})$*:* Dec *takes as input the public key* pk*, a ciphertext* c *and a secret key* sk*. The algorithm outputs a message* $m \in \mathcal{M}$ *when* $\mathbb{A}$ *accepts* S *and an error symbol* $\perp$ *otherwise.*

Next we define what it means for a CP-ABE scheme to be IND-CPA adaptively secure. The security experiment goes as follow:

The security experiment $\mathsf{Exp}^{\mathsf{CP\text{-}ABE}}_{\mathcal{A}}(\lambda)$

- **Initialization**: The challenger runs $\mathsf{Setup}(1^\lambda)$ to obtain $(\mathsf{pk}, \mathsf{msk})$ and gives pk to the adversary $\mathcal{A}$.
- **Key generation queries** $\mathsf{KeyGen}(\mathsf{msk}, S)$: The adversary $\mathcal{A}$ sends a set S of attributes. The challenger runs $\mathsf{KeyGen}(\mathsf{msk}, S)$ to obtain a key sk which is returned to $\mathcal{A}$. This step can be repeated as many times as $\mathcal{A}$ desires, with the condition that attribute queries S do not satisfy $\mathbb{A}^\star$.
- **Encryption queries** $\mathsf{Enc}(\mathbb{A}^\star, m_0, m_1)$: The adversary $\mathcal{A}$ submits a pair of messages m_0, m_1 and an access structure $\mathbb{A}^\star$ with the condition that no previously queried S satisfies $\mathbb{A}^\star$. The challenger flips a fair coin b and runs $\mathsf{Enc}(\mathsf{pk}, \mathbb{A}^\star, m_b)$ to obtain a ciphertext c, which is returned to $\mathcal{A}$.
- **Finalize**: The adversary $\mathcal{A}$ submits a guess b'. The output of the experiment is defined to be 1 if $b = b'$, and 0 otherwise.

Definition 7 (Security for CP-ABE Schemes) *The CP-ABE scheme is said to be* IND-CPA *adaptively secure, if for all PPT adversaries* $\mathcal{A}$ *a negligible function exists, such that*

$$\Pr\left[\mathsf{Exp}^{\mathsf{CP\text{-}ABE}}_{\mathcal{A}}(\lambda) = 1\right] \leq \frac{1}{2} + negl(\lambda).$$

3.2 Key-Policy Attribute-Based Encryption

Now we define KP-ABE. In CP-ABE the conditions to access a message are embedded in the ciphertext while access rights are embedded in a secret key. Here, access rights are put inside the ciphertext and access conditions are put into a secret key.

Definition 8 (KP-ABE Scheme [18]) *A KP-ABE scheme over a message space $\mathcal{M}$ is a tuple of four PPT algorithms* $\mathsf{CP-ABE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$*, such that*

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{msk})$*:* Setup *takes as input the security parameter λ. The algorithm outputs the public key* pk *and a master key* msk.
- $\mathsf{Enc}(\mathsf{pk}, S, m) \to c$*:* Enc *takes as inputs the public key* pk*, a set of attributes S and a message $m \in \mathcal{M}$. The algorithm outputs a ciphertext c.*
- $\mathsf{KeyGen}(\mathsf{msk}, \mathbb{A}) \to \mathsf{sk}$*:* KeyGen *takes as input the master key* msk *and an access structure $\mathbb{A}$. The algorithm outputs a secret key* sk.
- $\mathsf{Dec}(\mathsf{pk}, c, \mathsf{sk})$*:* Dec *takes as input the public key* pk*, a ciphertext c and a secret key* sk*. The algorithm outputs a message $m \in \mathcal{M}$ when $\mathbb{A}$ accepts S and an error symbol $\perp$ otherwise.*

Next we define what it means for a KP-ABE scheme to be IND-CPA adaptively secure. The security experiment goes as follow:

The security experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{KP\text{-}ABE}}(\lambda)$

- **Initialization**: The challenger runs $\mathsf{Setup}(1^\lambda)$ to obtain (p_k, m_k) and gives pk to the adversary $\mathcal{A}$.
- **Key generation queries** $\mathsf{KeyGen}(\mathsf{msk}, \mathbb{A})$: The adversary $\mathcal{A}$ sends an access tree $\mathbb{A}$ to the challenger. The challenger runs $\mathsf{KeyGen}(\mathsf{msk}, \mathbb{A})$ to obtain a key sk which is returned to $\mathcal{A}$. This step can be repeated as many times as $\mathcal{A}$ desires, with the condition that no access structures queried accept $S^\star$.
- **Encryption queries** $\mathsf{Enc}(S^\star, m_0, m_1)$: The adversary $\mathcal{A}$ submits a pair of messages m_0, m_1 and a set of attributes $S^\star$ with the condition that no previously queried $\mathbb{A}$ accepts $S^\star$. The challenger flips a fair coin b and runs $\mathsf{Enc}(\mathsf{pk}, S^\star, m_b)$ to obtain a ciphertext c, which is returned to $\mathcal{A}$.
- **Finalize**: The adversary $\mathcal{A}$ submits a guess b'. The output of the experiment is defined to be 1 if $b = b'$, and 0 otherwise.

Definition 9 (Security for KP-ABE Schemes) *The KP-ABE scheme is said to be* IND-CPA *adaptively secure, if for all PPT adversaries $\mathcal{A}$ a negligible function exists, such that*

$$\Pr\left[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{KP\text{-}ABE}}(\lambda) = 1\right] \leq \frac{1}{2} + negl(\lambda).$$

3.3 Multi-Client Functional Encryption

Multi-client functional encryption (MCFE) is a more flexible variant of functional encryption whose functional decryption involves multiple ciphertexts from different parties. Each party holds a different secret key and can independently and adaptively be corrupted by the adversary.

Now, we define a private-key $MCFE$ scheme as in [17]:

Definition 10 ***(Multi-Client Functional Encryption)*** *Let $\mathcal{F} = \{\mathcal{F}_\rho\}_\rho$ be a family (indexed by ρ) of sets $\mathcal{F}_\rho$ of functions $f : \mathcal{X}_{\rho,1} \times \cdots \times \mathcal{X}_{\rho,n_\rho} \to \mathcal{Y}_\rho$.*[1] *Let* $\mathsf{Labels} = \{0,1\}^*$ *or* $\{\perp\}$ *be a set of labels. A multi-client functional encryption scheme (MCFE) for the function family $\mathcal{F}$ and the label set* Labels *is a tuple of five algorithms* $\mathsf{MCFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$*:*

$\mathsf{Setup}(1^\lambda, 1^n)$**:** *Takes as input a security parameter λ and the number of parties n, and generates public parameters* pp. *The public parameters implicitly define an index ρ corresponding to a set $\mathcal{F}_\rho$ of n-ary functions (i.e., $n = n_\rho$).*

$\mathsf{KeyGen}(\mathsf{pp})$**:** *Takes as input the public parameters* pp *and outputs n secret keys* $\{\mathsf{sk}_i\}_{i\in[n]}$ *and a master secret key* msk.

$\mathsf{KeyDer}(\mathsf{pp}, \mathsf{msk}, f)$**:** *Takes as input the public parameters* pp*, the master secret key* msk *and a function $f \in \mathcal{F}_\rho$, and outputs a functional decryption key* sk_f.

$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, x_i, \ell)$**:** *Takes as input the public parameters* pp*, a secret key* sk_i*, a message $x_i \in \mathcal{X}_{\rho,i}$ to encrypt, a label $\ell \in$* Labels*, and outputs ciphertext* $\mathsf{ct}_{i,\ell}$.

$\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, \mathsf{ct}_{1,\ell}, \ldots, \mathsf{ct}_{n,\ell})$**:** *Takes as input the public parameters* pp*, a functional key* sk_f *and n ciphertexts under the same label ℓ and outputs a value $y \in \mathcal{Y}_\rho$.*

A scheme MCFE *is correct, if for all $\lambda, n \in \mathbb{N}$,* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$, $f \in \mathcal{F}_\rho$, $\ell \in \mathsf{Labels}$, $x_i \in \mathcal{X}_{\rho,i}$, *when* $(\{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{msk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$ *and* $\mathsf{sk}_f \leftarrow \mathsf{KeyDer}(\mathsf{pp}, \mathsf{msk}, f)$*, we have*

$$\Pr\left[\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_1, x_1, \ell), \ldots, \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_n, x_n, \ell)) = f(x_1, \ldots, x_n)\right] = 1 \ .$$

When ρ is clear from context, the index ρ is omitted. When $\mathsf{Labels} = \{0,1\}^*$, we say that the scheme is *labeled* or *with labels*. When $\mathsf{Labels} = \{\perp\}$, we say that the scheme is *without labels*, and we often omit ℓ.

The security model of multi-client functional encryption is similar to the security model of standard multi-input functional encryption, except that instead of a single master secret key msk for encryption, each slot i has a different secret key sk_i and the keys sk_i can be individually corrupted. In addition, one also needs to consider corruptions to handle possible collusions between different parties. In the following, we define security as adaptive left-or-right indistinguishability under both static (sta), and adaptive (adt) corruption. We also consider three variants of these notions (one, any, pos) related to the number of encryption queries asked by the adversary for each slot.

Definition 11 ***(Security of MCFE)*** *Let* MCFE *be an MCFE scheme, $\mathcal{F} = \{\mathcal{F}_\rho\}_\rho$ a function family indexed by ρ and* Labels *a label set. For* $\mathrm{xx} \in \{\mathrm{sta}, \mathrm{adt}\}$, $\mathrm{yy} \in \{\mathrm{one}, \mathrm{any}, \mathrm{pos}\}$*, and $\beta \in \{0,1\}$, we define the experiment* xx-yy-$\mathit{IND}_\beta^{\mathsf{MCFE}}$ *in Fig. 1, where the oracles are defined as:*

Corruption oracle $\mathsf{QCor}(i)$**:** *Outputs the encryption key* sk_i *of slot i. We denote by $\mathcal{CS}$ the set of corrupted slots at the end of the experiment.*

Encryption oracle $\mathsf{QEnc}(i, x_i^0, x_i^1, \ell)$**:** *Outputs* $\mathsf{ct}_{i,\ell} = \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, x_i^\beta, \ell)$ *on a query (i, x_i^0, x_i^1, ℓ). We denote by $Q_{i,\ell}$ the number of queries of the form* $\mathsf{QEnc}(i, \cdot, \cdot, \ell)$.

Key derivation oracle $\mathsf{QKeyD}(f)$**:** *Outputs* $\mathsf{sk}_f = \mathsf{KeyDer}(\mathsf{pp}, \mathsf{msk}, f)$.

and where Condition (*) *holds if all the following conditions hold:*

[1] All the functions inside the same set $\mathcal{F}_\rho$ have the same domain and the same range.

sta-yy-IND$_\beta^{\mathsf{MCFE}}(\lambda, n, \mathcal{A})$	**adt-yy-IND**$_\beta^{\mathsf{MCFE}}(\lambda, n, \mathcal{A})$
$\mathcal{CS} \leftarrow \mathcal{A}(1^\lambda, 1^n)$	$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$	$(\{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{msk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
$(\{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{msk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$	$\alpha \leftarrow \mathcal{A}^{\mathsf{QCor}(\cdot),\mathsf{QEnc}(\cdot,\cdot,\cdot,\cdot),\mathsf{QKeyD}(\cdot)}(\mathsf{pp})$
$\alpha \leftarrow \mathcal{A}^{\mathsf{QEnc}(\cdot,\cdot,\cdot,\cdot),\mathsf{QKeyD}(\cdot)}(\mathsf{pp}, \{\mathsf{sk}_i\}_{i\in\mathcal{CS}})$	**Output:** α if Condition (*) is satisfied, or a uniform bit otherwise
Output: α if Condition (*) is satisfied, or a uniform bit otherwise	

Figure 1: Security games for MCFE

- *If $i \in \mathcal{CS}$ (i.e., slot i is corrupted): for any query* $\mathsf{QEnc}(i, x_i^0, x_i^1, \ell)$, $x_i^0 = x_i^1$.
- *For any label $\ell \in$* Labels, *for any family of queries* $\{\mathsf{QEnc}(i, x_i^0, x_i^1, \ell)\}_{i\in[n]\setminus\mathcal{CS}}$, *for any family of inputs* $\{x_i \in \mathcal{X}_{\rho,i}\}_{i\in\mathcal{CS}}$, *for any query* $\mathsf{QKeyD}(f)$, *we define* $x_i^0 = x_i^1 = x_i$ *for any slot* $i \in \mathcal{CS}$, $\vec{x}^b = (x_1^b, \dots, x_n^b)$ *for* $b \in \{0, 1\}$, *and we require that:*
$$f(\vec{x}^0) = f(\vec{x}^1) \quad .$$
We insist that if one index $i \notin \mathcal{CS}$ is not queried for the label ℓ, there is no restriction.
- *When* $\mathrm{yy} = \mathrm{one}$*: for any slot $i \in [n]$ and $\ell \in$* Labels, $Q_{i,\ell} \in \{0, 1\}$, *and if* $Q_{i,\ell} = 1$, *then for any slot* $j \in [n] \setminus \mathcal{CS}$, $Q_{j,\ell} = 1$. *In other words, for any label, either the adversary makes no encryption query or makes exactly one encryption query for each* $i \in [n] \setminus \mathcal{CS}$.
- *When* $\mathrm{yy} = \mathrm{pos}$*: for any slot $i \in [n]$ and $\ell \in$* Labels, *if* $Q_{i,\ell} > 0$, *then for any slot* $j \in [n] \setminus \mathcal{CS}$, $Q_{j,\ell} > 0$. *In other words, for any label, either the adversary makes no encryption query or makes at least one encryption query for each slot* $i \in [n] \setminus \mathcal{CS}$.

We define the advantage of an adversary $\mathcal{A}$ in the following way:

$$\mathsf{Adv}^{\text{xx-yy-}\mathit{IND}}_{\mathsf{MCFE},\mathcal{A}}(\lambda, n) = \left|\Pr[\text{xx-yy-IND}_0^{\mathsf{MCFE}}(\lambda, n, \mathcal{A}) = 1] - \Pr[\text{xx-yy-IND}_1^{\mathsf{MCFE}}(\lambda, n, \mathcal{A}) = 1]\right| \ .$$

A multi-client functional encryption scheme MCFE *is* xx-yy-*IND secure, if for any n, for any polynomial-time adversary $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that:* $\mathsf{Adv}^{\text{xx-yy-}\mathit{IND}}_{\mathsf{MCFE},\mathcal{A}}(\lambda, n) \le \mathsf{negl}(\lambda)$.

We omit n when it is clear from the context. We also often omit $\mathcal{A}$ from the parameter of experiments or games when it is clear from context.

3.4 Decentralized Multi-Client Functional Encryption

Now, we recall the definition of decentralized multi-client functional encryption (DMCFE) [13]. As in the MCFE definition, the Setup algorithm, which generates the public parameters that determine the set of functions, is separated from the KeyGen algorithm.

Definition 12 ***(Decentralized Multi-Client Functional Encryption)*** *Let* $\mathcal{F} = \{\mathcal{F}_\rho\}_\rho$ *be a family (indexed by ρ) of sets $\mathcal{F}_\rho$ of functions* $f : \mathcal{X}_{\rho,1} \times \cdots \times \mathcal{X}_{\rho,n_\rho} \to \mathcal{Y}_\rho$*.Let* $\mathsf{Labels} = \{0, 1\}^*$ *or* $\{\perp\}$ *be a set of labels. A decentralized multi-client functional encryption scheme (DMCFE) for the function family $\mathcal{F}$ and the label set* Labels *is a tuple of six algorithms* $\mathsf{DMCFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{KeyDerShare}, \mathsf{KeyDerComb}, \mathsf{Enc}, \mathsf{Dec})$*:*

$\mathsf{Setup}(1^\lambda, 1^n)$ *is defined as for MCFE in Definition 10.*

$\mathsf{KeyGen}(\mathsf{pp})$**:** *Takes as input the public parameters* pp *and outputs* n *secret keys* $\{\mathsf{sk_i}\}_{i\in[n]}$.

$\mathsf{KeyDerShare}(\mathsf{pp}, \mathsf{sk}_i, f)$**:** *Takes as input the public parameters* pp*, a secret key* sk_i *from position* i *and a function* $f \in \mathcal{F}_\rho$*, and outputs a partial functional decryption key* $\mathsf{sk}_{i,f}$.

$\mathsf{KeyDerComb}(\mathsf{pp}, \mathsf{sk}_{1,f}, \ldots, \mathsf{sk}_{n,f})$**:** *Takes as input the public parameters* pp*,* n *partial functional decryption keys* $\mathsf{sk}_{1,f}, \ldots, \mathsf{sk}_{n,f}$ *and outputs the functional decryption key* sk_f.

$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, x_i, \ell)$ *is defined as for MCFE in Definition 10.*

$\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, \mathsf{ct}_{1,\ell}, \ldots, \mathsf{ct}_{n,\ell})$ *is defined as for MCFE in Definition 10.*

A scheme DMCFE *is correct, if for all* $\lambda, n \in \mathbb{N}$, $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$, $f \in \mathcal{F}_\rho$, $\ell \in \mathsf{Labels}$, $x_i \in \mathcal{X}_{\rho,i}$*, when* $\{\mathsf{sk}_i\}_{i\in[n]} \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, $\mathsf{sk}_{i,f} \leftarrow \mathsf{KeyDerShare}(\mathsf{sk}_i, f)$ *for* $i \in [n]$*, and* $\mathsf{sk}_f \leftarrow \mathsf{KeyDerComb}(\mathsf{pp}, \mathsf{sk}_{1,f}, \ldots, \mathsf{sk}_{n,f})$*, we have*

$$\Pr\left[\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_1, x_1, \ell), \ldots, \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_n, x_n, \ell)) = f(x_1, \ldots, x_n)\right] = 1 \ .$$

We remark that there is no master secret key msk. Furthermore, as in [13], the definition above does not explicitly ask the setup to be decentralized.

We consider a similar security definition for the decentralized multi-client scheme.

Definition 13 ***(Security of DMCFE)*** *The* xx-yy-*IND security notion of an DMCFE scheme (*$\mathrm{xx} \in \{\mathrm{sta}, \mathrm{adt}\}$ *and* $\mathrm{yy} \in \{\mathrm{one}, \mathrm{any}, \mathrm{pos}\}$*) is similar to the one of an MCFE (Definition 11), except that there is no master secret key* msk *and the key derivation oracle is now defined as:*

Key derivation oracle $\mathsf{QKeyD}(f)$**:** *Computes* $\mathsf{sk}_{i,f} := \mathsf{KeyDerShare}(\mathsf{pp}, \mathsf{sk}_i, f)$ *for* $i \in [n]$ *and outputs* $\{\mathsf{sk}_{i,f}\}_{i\in[n]}$.

3.5 Inner-Product Functionality

The inner-product functionality can be described by considering the index ρ of $\mathcal{F}$ in more detail.

The index of the family is defined as $\rho = (\mathcal{R}, n, m, X, Y)$ where $\mathcal{R}$ is either $\mathbb{Z}$ or $\mathbb{Z}_L$ for some integer L, and n, m, X, Y are positive integers. If X, Y are omitted, then $X = Y = L$ is used (i.e., no constraint).

This defines $\mathcal{F}_\rho = \{f_{\vec{y}_1,\ldots,\vec{y}_n} : (\mathcal{R}^m)^n \to \mathcal{R}\}$ where

$$f_{\vec{y}_1,\ldots,\vec{y}_n}(\vec{x}_1, \ldots, \vec{x}_n) = \sum_{i=1}^{n} \langle \vec{x}_i, \vec{y}_i \rangle = \langle \vec{x}, \vec{y} \rangle \ ,$$

where the vectors satisfy the following bounds: $\|\vec{x}_i\|_\infty < X$, $\|\vec{y}_i\|_\infty < Y$ for $i \in [n]$, and where $\vec{x} \in \mathcal{R}^{mn}$ and $\vec{y} \in \mathcal{R}^{mn}$ are the vectors corresponding to the concatenation of the n vectors $\vec{x}_1, \ldots, \vec{x}_n$ and $\vec{y}_1, \ldots, \vec{y}_n$ respectively.

3.6 Symmetric Encryption

One of the compilers in Section 5 makes use of a symmetric encryption scheme $\mathsf{SE} = (\mathsf{Enc_{SE}}, \mathsf{Dec_{SE}})$ that is indistinguishable secure under chosen-plaintext attacks (IND-CPA) and whose keys are uniform strings in $\{0,1\}^\lambda$ as defined by [10].

$\mathsf{Enc}_{\mathsf{SE}}(\mathsf{K}, x)$**:** Takes as input a key $\mathsf{K} \in \{0,1\}^{\lambda}$ and a message x to encrypt, and outputs the ciphertext ct.

$\mathsf{Dec}_{\mathsf{SE}}(\mathsf{K}, \mathsf{ct})$**:** Takes as input a key K and a ciphertext ct to decrypt, and outputs a message x.

We denote with $\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{SE},\mathcal{A}}(\lambda)$ the advantage of an adversary guessing β in the following game: the challenger picks $\mathsf{K} \leftarrow \{0,1\}^{\lambda}$ and $\beta \leftarrow \{0,1\}$ and gives $\mathcal{A}$ access to an encryption oracle $\mathsf{QEnc}(x_i^0, x_i^1)$ that outputs $\mathsf{ct} = \mathsf{Enc}_{\mathsf{SE}}(\mathsf{K}, x_i^{\beta})$ on a query (x^0, x^1).

4 Digital Currency Scenario

The following is dedicated to presenting FAME [6]. This pair of schemes, for CP and KP ABE, was already our choice in year 1 of the project. At the time of D4.1 and D4.2, our only concern was that the scheme capabilities for attribute sets may be too limiting for our use case. With recent developments and the progress of the prototype, this is no longer a concern, and so we confirm that FAME is the best solution available.

4.1 Preliminary

4.1.1 Access Structure and Access Control

Definition 14 (Access Structure) *If $\mathcal{U}$ denotes the universe of attributes, then an access structure $\mathbb{A}$ is a collection of non-empty subsets of $\mathcal{U}$, i.e, $\mathcal{A} \subseteq 2^{\mathcal{U}} \backslash \{\emptyset\}$, where $2^{\mathcal{U}} = \{X, X \subseteq \mathcal{U}\}$. It is called monotone if for every $B, C \subseteq \mathcal{U}$ such that $B \subseteq C$, $B \in \mathbb{A} \implies C \in \mathbb{A}$.*

Remark 1 *Monotony captures the idea that authorized users having attributes B happening to get more attributes $C \supseteq B$ do not lose privileges: $B \in \mathbb{A} \implies C \in \mathbb{A}$.*

Access control can be seen as monotone boolean formulae with AND and OR gates, where each input is associated with an attribute in $\mathcal{U}$. A set of attributes $S \subseteq \mathcal{U}$ satisfies a formula if it evaluates to true on setting all inputs that map to some attribute in S to true, and the rest to false. Boolean formulae fall into a more general class of functions called Monotone Span Programs (MSP), defined in [18] in the context of ABE schemes and recalled in [6].

Definition 15 (Monotone Span Program [18, 6]) *An MSP is given by a matrix M of size $n_1 \times n_2$ over $\mathbb{Z}_p$ and a mapping $\pi : \{1, ..., n_1\} \to \mathcal{U}$. Let S be a set of attributes and $I = \{i, i \in \{1, \cdots n_1\}, \pi(i) \in S\}$ be the set of rows in M that belongs to S. We say that M accepts S if there exist a linear combination of rows in I that gives $(1, 0, \cdots, 0)$. More formally, there should exist coefficients $\{\gamma_i\}_{i \in I}$ such that*

$$\sum_{i \in I} \gamma_i (M)_i = (1, 0, \cdots, 0), \tag{1}$$

where $(M)_i$ is the ith row of M.

4.1.2 Hash Function

The FAME scheme uses a collision-resistant hash function that we will call $\mathcal{H}$. This hash function $\mathcal{H}$ takes two types of inputs, one type of the form (x, l, t) and one type of the form (j, l, t), where x is an arbitrary string, j is a positive integer, $l \in \{1, 2, 3\}$ and $t \in \{1, 2\}$. For simplicity, we represent these two inputs as xlt and $0jlt$ respectively.

4.2 An instantiation of Ciphertext-Policy Attribute-Based Encryption

In this section, we give a formal description of FAME, a *CP-ABE* scheme presented in [6], as well as a security statement for it. For a complete security proof, please refer to the original article [6].

The following protocol makes use of a PPT pairing group generator algorithm *GroupGen* that on input 1^λ returns a description $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ of asymmetric pairing groups where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are cyclic groups of order p for a 2λ-bit prime p, g_1 and g_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ is an efficiently computable (non-degenerate) bilinear map, as described in Section 2.3.

In the actual description of the protocol, in order to be consistent with the description in [6], the values $\mathbb{G}_1$, $\mathbb{G}_2$, g_1, and g_2 are denoted by $\mathbb{G}$, $\mathbb{H}$, g, and h, respectively. Moreover, $GroupGen(1^\lambda)$ is assumed to be an algorithm outputting a Type-3 pairing curve, for which there are no efficiently computable homomorphisms between $\mathbb{G}$ and $\mathbb{H}$ [16].

FAME, an instantiation of a CP-ABE Scheme [6]

- $\underline{\mathsf{Setup}(1^\lambda)}$ Run $GroupGen(1^\lambda)$ to obtain $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, p, g, h, e)$. Then sample $a_1, a_2 \leftarrow_{\$} \mathbb{Z}_p^*$ and $d_1, d_2, d_3 \leftarrow_{\$} \mathbb{Z}_p$. Output

$$(h, H_1 = h^{a_1}, H_2 = h^{a_2}, T_1 = e(g,h)^{d_1 a_1 + d_3}, T_2 = e(g,h)^{d_2 a_2 + d_3})$$

as the public key. Also, sample $b_1, b_2 \leftarrow_{\$} \mathbb{Z}_p^*$ and output the master key

$$(g, h, a_1, a_2, b_1, b_2, g^{d_1}, g^{d_2}, g^{d_3}).$$

- $\underline{\mathsf{Enc}(\mathsf{pk}, (M, \pi), m)}$ Sample $s_1, s_2 \leftarrow_{\$} \mathbb{Z}_p$. Compute

$$c_0 := (H_1^{s_1}, H_2^{s_2}, h^{s_1+s_2})$$

using pk. Suppose M has n_1 rows and n_2 columns. Then, for $i = 1, \cdots, n_1$ and $l = 1, 2, 3$, compute

$$c_{i,l} = \mathcal{H}(\pi(i)l1)^{s_1} \cdot \mathcal{H}(\pi(i)l2)^{s_2} \cdot \prod_{j=1}^{n_2} [\mathcal{H}(0jl1)^{s_1} \cdot \mathcal{H}(0jl2)^{s_2}]^{(M)_{i,j}},$$

where $(M)_{i,j}$ denotes the (i, j)th element of M. Set $c_i := (c_{i,1}, c_{i,2}, c_{i,3})$. Also, compute

$$c' = T_1^{s_1} \cdot T_2^{s_2} \cdot m.$$

Output $(c_0, c_1, \cdots, c_{n_1}, c')$ as the ciphertext.

- $\underline{\mathsf{KeyGen}(\mathsf{msk}, S)}$ Sample $r_1, r_2 \leftarrow_{\$} \mathbb{Z}_p$ and compute

$$\mathsf{sk}_0 := (h^{b_1 r_1}, h^{b_2 r_2}, h^{r_1+r_2})$$

using h, b_1, b_2 from mk. For all $y \in S$ and $t = 1, 2$, compute

$$\mathsf{sk}_{y,t} = \mathcal{H}(y1t)^{\frac{b_1 r_1}{a_t}} \cdot \mathcal{H}(y2t)^{\frac{b_2 r_2}{a_t}} \cdot \mathcal{H}(y3t)^{\frac{\sigma_y}{a_t}} \cdot g^{\frac{\sigma'}{a_t}},$$

where $\sigma_y \leftarrow_{\$} \mathbb{Z}_p$. Set $\mathsf{sk}_y := (\mathsf{sk}_{y,1}, \mathsf{sk}_{y,2}, g^{-\sigma_y})$. Also, compute

$$\mathsf{sk}'_t = g^{d_t} \cdot \mathcal{H}(011t)^{\frac{b_1 r_1}{a_t}} \cdot \mathcal{H}(012t)^{\frac{b_2 r_2}{a_t}} \cdot \mathcal{H}(013t)^{\frac{r_1+r_2}{a_t}} \cdot g^{\frac{\sigma'}{a_t}}$$

for $t = 1, 2$, where $\sigma' \leftarrow_{\$} \mathbb{Z}_p$. Set $\mathsf{sk}' := (\mathsf{sk}'_1, \mathsf{sk}'_2, g^{d_3} \cdot g^{-\sigma'})$. Output $(\mathsf{sk}_0, \{\mathsf{sk}_y\}_{y \in S}, sk')$ as the key.

- Dec(pk, c, sk) Recall that if the S used for generating sk satisfies the MSP (M, π) used for encrypting c, then there exist $\{\gamma_i\}_{i \in I}$ satisfying Eq. (1). Now, compute

$$\begin{aligned} num &:= c' \cdot e\left(\prod_{i \in I} c_{i,1}^{\gamma_i}, \mathsf{sk}_{0,1}\right) \cdot e\left(\prod_{i \in I} c_{i,2}^{\gamma_i}, \mathsf{sk}_{0,2}\right) \cdot e\left(\prod_{i \in I} c_{i,3}^{\gamma_i}, \mathsf{sk}_{0,3}\right), \\ den &:= e\left(\mathsf{sk}_1' \cdot \prod_{i \in I} \mathsf{sk}_{\pi(i),1}^{\gamma_i}, c_{0,1}\right) \cdot e\left(\mathsf{sk}_2' \cdot \prod_{i \in I} \mathsf{sk}_{\pi(i),2}^{\gamma_i}, c_{0,2}\right) \cdot e\left(\mathsf{sk}_3' \cdot \prod_{i \in I} \mathsf{sk}_{\pi(i),3}^{\gamma_i}, c_{0,3}\right), \end{aligned}$$

  and output num/dem. Here $\mathsf{sk}_{0,1}$,$\mathsf{sk}_{0,2}$,$\mathsf{sk}_{0,3}$ denote the first, second and third elements of sk_0; the same for c_0.

4.2.1 Security

The scheme is proven to be IND-CPA adaptively secure (Definition 6) in the Random-Oracle Model (ROM) under the Decisional LINear (DLIN) assumption, which is a particular case of the Matrix Diffie-Hellman assumption (Definition 4) which is a classical hardness assumption. The proof for the following theorem is given in [6].

Theorem 1 *FAME is adaptively secure under the DLIN assumption on asymmetric pairing groups in the random oracle model. Concretely, for any PPT adversary $\mathcal{A}$ making Q key queries in the* IND-CPA *security game, there exists a PPT adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}_{FAME}^{\mathcal{A}}(\lambda) \leq (8Q + 2)\mathsf{Adv}_{DLIN}^{\mathcal{B}}(\lambda) + (16Q + 6)/p,$$

where $p = \Theta(\lambda)$ is the order of the pairing group.

4.3 An instantiation of Key-Policy Attribute-Based Encryption

The KP-ABE scheme we present here is derived from FAME. As such it enjoys the same efficiency and expressiveness as FAME.

An instantiation of a KP-ABE Scheme [6]

- Setup(1^λ) Same Setup() as FAME.

- Enc(pk, S, m) Sample $s_1, s_2 \leftarrow_{\$} \mathbb{Z}_p$. Compute

$$c_0 := (H_1^{s_1}, H_2^{s_2}, h^{s_1+s_2})$$

  using pk. For all $y \in S$ and $l = 1, 2, 3$, compute

$$c_{y,l} := \mathcal{H}(yl1)^{s_1} \cdot \mathcal{H}(yl2)^{s_2}.$$

  Set $c_y := (c_{y,1}, c_{y_2}, c_{y,3})$. Also, compute

$$c' := T_1^{s_1} \cdot T_2^{s_2} \cdot m.$$

  Output $(c_0, \{c_y\}_{y \in S}, c')$ as the ciphertext.

- KeyGen(msk, (M, π)) Sample $r_1, r_2 \leftarrow_{\$} \mathbb{Z}_p$ and compute

$$\mathsf{sk}_0 := (h^{b_1 r_1}, h^{b_2 r_2}, h^{r_1+r_2})$$

  using h, b_1, b_2 from msk. Sample $\sigma'_2, \cdots, \sigma'_{n_2} \leftarrow_{\$} \mathbb{Z}_p$. For all $i = 1, \cdots, n_1$ and $t = 1, 2$, compute

$$\begin{aligned} \mathsf{sk}_{i,t} &:= \mathcal{H}(\pi(i)1t)^{\frac{b_1 r_1}{a_t}} \cdot \mathcal{H}(\pi(i)2t)^{\frac{b_2 r_2}{a_t}} \cdot \mathcal{H}(\pi(i)3t)^{\frac{r_1+r_2}{a_t}} \cdot g^{\frac{\sigma_i}{a_t}} \cdot (g^{d_t})^{(M)_{i,1}} \cdot \\ &\prod_{j=2}^{n_2} \left[\mathcal{H}(0j1t)^{\frac{b_1 r_1}{a_t}} \cdot \mathcal{H}(0j2t)^{\frac{b_2 r_2}{a_t}} \cdot \mathcal{H}(0j3t)^{\frac{r_1+r_2}{a_t}} \right]^{(M)_{i,1}}, \\ \mathsf{sk}_{i,3} &:= g^{-\sigma_i} \cdot (g^{d_3})^{(M)_{i,1}} \cdot \prod_{j=2}^{n_2} \left(g^{-\sigma'_j} \right)^{(M)_{i,j}}, \end{aligned}$$

  where $\sigma_i \leftarrow_{\$} \mathbb{Z}_p$. Set $\mathsf{sk}_i := (\mathsf{sk}_{i,1}, \mathsf{sk}_{i,2}, \mathsf{sk}_{i,3})$. Output $(\mathsf{sk}_0, \mathsf{sk}_1, \cdots, \mathsf{sk}_{n_1})$ as the key.

- Dec(pk, c, sk) Recall that if the S used for generating sk satisfies the MSP (M, π) used for encrypting c, then there exist $\{\gamma_i\}_{i \in I}$ satisfying Eq. (1). Now, compute

$$\begin{aligned} num &:= c' \cdot e\left(\prod_{i \in I} c_{\pi(i),1}^{\gamma_i}, \mathsf{sk}_{0,1} \right) \cdot e\left(\prod_{i \in I} c_{\pi(i),2}^{\gamma_i}, \mathsf{sk}_{0,2} \right) \cdot e\left(\prod_{i \in I} c_{\pi(i),3}^{\gamma_i}, \mathsf{sk}_{0,3} \right), \\ den &:= e\left(\cdot \prod_{i \in I} \mathsf{sk}_{i,1}^{\gamma_i}, c_{0,1} \right) \cdot e\left(\cdot \prod_{i \in I} \mathsf{sk}_{i,2}^{\gamma_i}, c_{0,2} \right) \cdot e\left(\cdot \prod_{i \in I} \mathsf{sk}_{i,3}^{\gamma_i}, c_{0,3} \right), \end{aligned}$$

  and output num/dem. Here $\mathsf{sk}_{0,1}$,$\mathsf{sk}_{0,2}$,$\mathsf{sk}_{0,3}$ denote the first, second and third elements of sk_0; the same for c_0.

4.3.1 Security Analysis

The security provided by this KP-ABE scheme is the same as the security provided by FAME.

5 Web Analytics Scenario

Recall that the objective of this use case is to create a scheme allowing some entities to participate in a privacy-preserving poll. Trust is central in this use-case. In the previous iteration of this document, we presented a compiler allowing to build a DMCFE scheme from a MCFE scheme. Decentralization is essential because it removes the need for a trusted entity, and this compiler meant that now, all MCFE schemes could potentially be a viable solution for the use case. Along with this compiler, we proposed a MCFE scheme under the DDH assumption that handles labels, as well as an MCFE scheme not handling labels but that could be instantiated with a wide variety of hardness assumptions. However, our DMCFE with labels DDH-based instantiation had two limitations; it was only able to decrypt for small messages and did not achieve post-quantum security. Along with the solutions already presented in D4.2, we present a new MCFE scheme with labels based on lattices [1] and developed during the project, that solves the issue we had in D4.2 by achieving post-quantum security and handling any size of decryption.

5.1 A Compiler from MCFE to Decentralized-MCFE

The following compiler from [2], developed in the context of FENTEC, turns MCFE schemes into DMCFE schemes. This compiler is property-preserving, meaning that if the starting MCFE scheme is static or adaptive secure, resistant to certain kinds of corruption or handle labels then the resulting DMCFE scheme will have the same properties. Since one of the MCFE schemes presented later is adaptive secure, resistant to any kind of corruption and can handle labels, the compiler yields a DMCFE scheme with all those properties.

```
$\mathsf{Setup}'(1^\lambda, 1^n)$ :
Return $\mathsf{Setup}(1^\lambda, 1^n)$

$\mathsf{KeyGen}'(\mathsf{pp})$ :
$(\{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{msk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
Recall that $\mathsf{sk}_i = (i, s_i, \{\vec{u}_i^k\}_{k\in[\kappa]})$
For $k \in [\kappa]$:
    For $i \in [n-1]$, $\vec{v}_i^k \leftarrow \mathbb{Z}_L^M$
    $\vec{v}_n^k := -\sum_{i=1}^{n-1} \vec{v}_i^k \bmod L$
Return $\{\mathsf{sk}'_i = (\mathsf{sk}_i, \{\vec{v}_i^k\}_{k\in[\kappa]})\}_{i\in[n]}$

$\mathsf{Enc}'(\mathsf{pp}, \mathsf{sk}'_i, x_i, \ell)$ :
Parse $\mathsf{sk}'_i = (\mathsf{sk}_i, \{\vec{v}_i^k\}_{k\in[\kappa]})$
Return $\mathsf{ct}_{i,\ell} \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, x_i, \ell)$

$\mathsf{KeyDerShare}'(\mathsf{pp}, \mathsf{sk}'_i, f)$ :
Parse $\mathsf{sk}'_i = (\mathsf{sk}_i, \{\vec{v}_i^k\}_{k\in[\kappa]})$
For $k \in [\kappa]$, $\mathsf{dk}_{i,f}^k := \langle \vec{u}_i^k, \vec{y}_{i,f}^k \rangle + \langle \vec{v}_i^k, \vec{y}_f^k \rangle$
Return $\mathsf{sk}'_{i,f} := (s_{i,f}, \{\mathsf{dk}_{i,f}^k\}_{k\in[\kappa]})$

$\mathsf{KeyDerComb}'(\mathsf{pp}, \{\mathsf{sk}'_{i,f}\}_{i\in[n]})$ :
Parse $\{\mathsf{sk}'_{i,f} = (s_{i,f}, \{\mathsf{dk}_{i,f}^k\}_{k\in[\kappa]})\}_{i\in[n]}$
For $k \in [\kappa]$, $\mathsf{dk}_f^k := \sum_{i=1}^{n} \mathsf{dk}_{i,f}^k$
Return $\mathsf{sk}'_f = (\{s_{i,f}\}_{i\in[n]}, \{\mathsf{dk}_f^k\}_{k\in[\kappa]})$

$\mathsf{Dec}'(\mathsf{pp}, \mathsf{sk}'_f, \{\mathsf{ct}_{i,\ell}\}_{i\in[n]})$ :
Return $\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}'_f, \{\mathsf{ct}_{i,\ell}\}_{i\in[n]})$
```

Figure 2: Compiler from MCFE to DMCFE′: $s_{i,f}$ is a function of pp, i, s_i, f and $\vec{y}_{i,f}^k$ is a function of pp, i, f, and k. $M = mn$.

5.1.1 Security Analysis

The following theorem, whose proof can be found in [2], shows that the resulting DMCFE scheme has the same security properties as the underlying MCFE scheme.

Theorem 2 *Let* $\mathsf{MCFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ *be an MCFE construction for a family of functions* $\mathcal{F}$ *and a set of labels* Labels. *We suppose that* MCFE *has the special key derivation property modulo a prime L. For any* $\mathrm{xx} \in \{\mathrm{sta}, \mathrm{adt}\}$ *and any* $\mathrm{yy} \in \{\mathrm{one}, \mathrm{pos}, \mathrm{any}\}$, *if* MCFE *is an* xx-yy-*IND-secure MCFE scheme, then the scheme* DMCFE' *depicted in Fig. 2 is an* xx-yy-*IND-secure DMCFE scheme. Namely, for any PPT adversary* $\mathcal{A}$, *there exist a PPT adversary* $\mathcal{B}$ *such that:*

$$\mathsf{Adv}^{\text{xx-yy-IND}}_{\mathsf{DMCFE}',\mathcal{A}}(\lambda, n) \leq \mathsf{Adv}^{\text{xx-yy-IND}}_{\mathsf{MCFE},\mathcal{B}}(\lambda, n) \ .$$

5.2 Security with Respect to Partial Queries

In this section, we present two compilers from [2], developed in the context of FENTEC, transforming pos-IND-secure MCFE and DMCFE schemes into any-IND schemes. These compilers essentially force the adversary to ask for at least one ciphertext per position i (and per label, for labeled schemes). The first compiler works for sta-pos-IND and adt-pos-IND-secure schemes without labels ($\mathsf{Labels} = \{\perp\}$) and only requires an IND-CPA symmetric encryption scheme to work. We prove it for the adt-pos-IND case as the proof for sta-pos-IND is simpler. The second compiler supports labeled schemes, but is in the random oracle model. Although our presentation is for DMCFE, the compilers can be adapted to work for MCFE schemes in a straightforward way.

```
$\mathsf{Setup}'(1^\lambda, 1^n)$ :
Return $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$

$\mathsf{KeyGen}'(\mathsf{pp})$ :
$\{\mathsf{sk}_i\}_{i \in [n]} \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
For $i \in [n]$ :
    $\mathsf{k}_{i,1}, \ldots, \mathsf{k}_{i,n} \leftarrow \{0,1\}^\lambda$
    $\mathsf{K}_i = \oplus_{j \in [n]} \mathsf{k}_{i,j}$
Return $\{\mathsf{sk}'_i = (\mathsf{sk}_i, \mathsf{K}_i, \{\mathsf{k}_{i,j}, \mathsf{k}_{j,i}\}_{j \in [n]})\}_{i \in [n]}$

$\mathsf{Enc}'(\mathsf{pp}, \mathsf{sk}'_i, x_i)$ :
Parse $\mathsf{sk}'_i = (\mathsf{sk}_i, \mathsf{K}_i, \{\mathsf{k}_{i,j}, \mathsf{k}_{j,i}\}_{j \in [n]})$
$\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, x_i)$
$\mathsf{ct}'_i \leftarrow \mathsf{Enc}_{\mathsf{SE}}(\mathsf{K}_i, \mathsf{ct}_i)$
Return $(\mathsf{ct}'_i, \{\mathsf{k}_{j,i}\}_{j \in [n]})$

$\mathsf{KeyDerShare}'(\mathsf{pp}, \mathsf{sk}'_i, f)$ :
Parse $\mathsf{sk}'_i = (\mathsf{sk}_i, \mathsf{K}_i, \{\mathsf{k}_{i,j}, \mathsf{k}_{j,i}\}_{j \in [n]})$
Return $\mathsf{sk}'_{i,f} \leftarrow \mathsf{KeyDerShare}(\mathsf{sk}_i, f)$

$\mathsf{KeyDerComb}'(\mathsf{pp}, \{\mathsf{sk}'_{i,f}\}_{i \in [n]})$ :
$\mathsf{sk}_f := \mathsf{KeyDerComb}(\mathsf{pp}, \{\mathsf{sk}'_{i,f}\}_{i \in [n]})$
Return $\mathsf{sk}_f$

$\mathsf{Dec}'(\mathsf{pp}, \mathsf{sk}_f, \mathsf{ct}''_1, \ldots, \mathsf{ct}''_n)$ :
Parse $\{\mathsf{ct}''_i = (\mathsf{ct}'_i, \{\mathsf{k}_{j,i}\}_{j \in [n]})\}_{i \in [n]}$
For $i \in [n]$ :
    $\mathsf{K}_i = \oplus_{j \in [n]} \mathsf{k}_{i,j}$
    $\mathsf{ct}_i \leftarrow \mathsf{Dec}_{\mathsf{SE}}(\mathsf{K}_i, \mathsf{ct}'_i)$
Return $\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$.
```

Figure 3: Compiler from an xx-pos-IND DMCFE DMCFE without labels into an xx-any-IND DMCFE DMCFE' using an IND-CPA symmetric-key encryption scheme SE

The compiler without labels is described in Fig. 3, where SE is an IND-CPA symmetric-key encryption scheme. The proof for the following security theorem can be found in [2].

Theorem 3 *Let* $\mathsf{DMCFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{KeyDerShare}, \mathsf{KeyDerComb}, \mathsf{Enc}, \mathsf{Dec})$ *be an* adt-pos-*IND-secure DMCFE scheme without labels* ($\mathsf{Labels} = \{\perp\}$) *for a family of functions* $\mathcal{F}$*. Let* $\mathsf{SE} = (\mathsf{Enc}_{\mathsf{SE}}, \mathsf{Dec}_{\mathsf{SE}})$ *be an IND-CPA symmetric-key encryption scheme. Then the DMCFE scheme* $\mathsf{DMCFE}' = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{KeyDerShare}', \mathsf{KeyDerComb}', \mathsf{Enc}', \mathsf{Dec}')$ *described in Fig. 3 is an* adt-any-*IND-secure DMCFE scheme. Namely, for any PPT adversary* $\mathcal{A}$*, there exist PPT adversaries* $\mathcal{B}$ *and* $\mathcal{B}'$ *such that:*

$$\mathsf{Adv}^{\text{adt-any-IND}}_{\mathsf{DMCFE}',\mathcal{A}}(\lambda, n) \leq \mathsf{Adv}^{\text{adt-pos-IND}}_{\mathsf{DMCFE},\mathcal{B}}(\lambda, n) + n \cdot \mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{SE},\mathcal{B}'}(\lambda) \enspace .$$

```
$\mathsf{Setup}'(1^\lambda, 1^n)$ :
Return $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$
$\mathsf{KeyGen}'(\mathsf{pp})$ :
$\{\mathsf{sk}_i\}_{i\in[n]} \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
For $i \in [n]$ :
    $\mathsf{k}_{i,1}, \ldots, \mathsf{k}_{i,n} \leftarrow \{0,1\}^\lambda$
Return $\{\mathsf{sk}'_i = (\mathsf{sk}_i, \mathsf{K}_i, \{\mathsf{k}_{i,j}, \mathsf{k}_{j,i}\}_{j\in[n]})\}_{i\in[n]}$

$\mathsf{Enc}'(\mathsf{pp}, \mathsf{sk}'_i, x_i, \ell)$ :
Parse $\mathsf{sk}'_i = (\mathsf{sk}_i, \mathsf{K}_i, \{\mathsf{k}_{i,j}, \mathsf{k}_{j,i}\}_{j\in[n]})$
$\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, x_i)$
For $j \in [n]$ :
    $\mathsf{k}_{i,j,\ell} := \mathsf{H}_1(\mathsf{k}_{i,j} \| i \| j \| \ell)$
    $\mathsf{k}_{j,i,\ell} := \mathsf{H}_1(\mathsf{k}_{j,i} \| j \| i \| \ell)$
$\mathsf{K}_{i,\ell} := \oplus_{j\in[n]} \mathsf{k}_{i,j,\ell}$
$r_i \leftarrow \{0,1\}^\lambda$; $\mathsf{ct}'_i := \mathsf{ct}_i \oplus \mathsf{H}_2(\mathsf{K}_{i,\ell} \| r_i)$
Return $(\mathsf{ct}'_i, r_i, \{\mathsf{k}_{j,i,\ell}\}_{j\in[n]})$
```

```
$\mathsf{KeyDerShare}'(\mathsf{pp}, \mathsf{sk}'_i, f)$ :
Parse $\mathsf{sk}'_i = (\mathsf{sk}_i, \mathsf{K}_i, \{\mathsf{k}_{i,j}, \mathsf{k}_{j,i}\}_{j\in[n]})$
Return $\mathsf{sk}'_{i,f} \leftarrow \mathsf{KeyDerShare}(\mathsf{pp}, \mathsf{sk}'_i, f)$

$\mathsf{KeyDerComb}'(\mathsf{pp}, \{\mathsf{sk}'_{i,f}\}_{i\in[n]})$ :
$\mathsf{sk}_f := \mathsf{KeyDerComb}(\mathsf{pp}, \{\mathsf{sk}_{i,f}\}_{i\in[n]})$
Return $\mathsf{sk}_f$

$\mathsf{Dec}'(\mathsf{pp}, \mathsf{sk}_f, \mathsf{ct}''_1, \ldots, \mathsf{ct}''_n)$ :
Parse $\{\mathsf{ct}''_i = (\mathsf{ct}'_i, r_i, \{\mathsf{k}_{j,i,\ell}\}_{j\in[n]})\}_{i\in[n]}$
For $i \in [n]$ :
    $\mathsf{K}_{i,\ell} = \oplus_{j\in[n]} \mathsf{k}_{i,j,\ell}$
    $\mathsf{ct}_i = \mathsf{ct}'_i \oplus \mathsf{H}_2(\mathsf{K}_{i,\ell} \| r_i)$
Return $\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_f, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$.
```

Figure 4: Compiler from an xx-pos-IND DMCFE DMCFE with labels into an xx-any-IND DMCFE DMCFE′ with labels, where $\mathsf{H}_1 : \{0,1\}^* \to \{0,1\}^\lambda$ and $\mathsf{H}_2 : \{0,1\}^* \to \{0,1\}^{|\mathsf{ct}_i|}$ are two hash functions modeled as random oracles in the security proof.

The compiler supporting labels is described in Fig. 4, where $\mathsf{H}_1 : \{0,1\}^* \to \{0,1\}^\lambda$ and $\mathsf{H}_2 : \{0,1\}^* \to \{0,1\}^{|\mathsf{ct}_i|}$ are two hash functions modeled as random oracles in the security proof. The proof for the following security theorem can be found in [2].

Theorem 4 *Let* $\mathsf{DMCFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{KeyDerShare}, \mathsf{KeyDerComb}, \mathsf{Enc}, \mathsf{Dec})$ *be an* adt-pos-*IND-secure DMCFE scheme for an ensemble of functions* $\mathcal{F}$ *and set of labels* Labels*. Then the DMCFE scheme* $\mathsf{DMCFE}' = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{KeyDerShare}', \mathsf{KeyDerComb}', \mathsf{Enc}', \mathsf{Dec}')$ *described in Fig. 4 is an* adt-any-*IND-secure scheme. Namely, when the hash functions* H_1 *and* H_2 *are modeled as random oracles, for any PPT adversary* $\mathcal{A}$ *there exist a PPT adversary* $\mathcal{B}$ *such that:*

$$\mathsf{Adv}^{\text{adt-any-IND}}_{\mathsf{DMCFE}',\mathcal{A}}(\lambda, n) \leq \mathsf{Adv}^{\text{adt-pos-IND}}_{\mathsf{DMCFE},\mathcal{B}}(\lambda, n) + \frac{2q_{\mathsf{H}_1} + (2n+1) \cdot (q_{\mathsf{H}_2} q_{\mathsf{QEnc}} + q^2_{\mathsf{QEnc}})}{2^\lambda} \,,$$

where q_{H_1}*,* q_{H_2}*, and* q_{QEnc} *are the numbers of queries to the oracles* H_1*,* H_2*, and* QEnc *respectively.*

5.3 A Multi-Client Functional Encryption Instantiation with Labels from DDH

The following MCFE scheme, from [12], is based on the DDH assumption. This scheme achieves adaptive security under any type of corruption (static and adaptive). It also handles labels, which allows more flexibility in practice. Along with the property-preserving compiler we presented earlier, it allows us to create our final DMCFE scheme.

- $\mathsf{Setup}(\lambda)$: Takes as input the security parameter, and generates prime-order group $\vec{\mathbb{G}} := (\mathbb{G}, p, P) \overset{R}{\leftarrow} \mathsf{GGen}(1^\lambda)$, and $\mathcal{H}$ a full-domain hash function onto $\mathbb{G}^2$. It also generates the encryption keys $\vec{s}_i \overset{R}{\leftarrow} \mathbb{Z}_p^2$, for $i = 1, \ldots, n$. The public parameters pk consist of $(\mathbb{G}, p, g, \mathcal{H})$, while the encryption keys are $\mathsf{ek}_i = \vec{s}_i$ for $i = 1, \ldots, n$, and the master secret key is $\mathsf{msk} = ((\mathsf{ek}_i)_i)$, (in addition to pk, which is omitted);
- $\mathsf{Encrypt}(\mathsf{ek}_i, x_i, \ell)$: Takes as input the value x_i to encrypt, under the key $\mathsf{ek}_i = \vec{s}_i$ and the label ℓ. It computes $[\vec{u}_\ell] := \mathcal{H}(\ell) \in \mathbb{G}^2$, and outputs the ciphertext $[c_i] = [\vec{u}_\ell^\top \vec{s}_i + x_i] \in \mathbb{G}$;
- $\mathsf{DKeyGen}(\mathsf{msk}, \vec{y})$: Takes as input $\mathsf{msk} = (\vec{s}_i)_i$ and an inner-product function defined by $\vec{y}$ as $f_{\vec{y}}(\vec{x}) = \mathsf{ip}\vec{x}\vec{y}$, and outputs the functional decryption key $\mathsf{dk}_{\vec{y}} = (\vec{y}, \sum_i \vec{s}_i \cdot y_i) \in \mathbb{Z}_p^n \times \mathbb{Z}_p^2$;
- $\mathsf{Decrypt}(\mathsf{dk}_{\vec{y}}, \ell, ([c_i])_{i \in [n]})$: Takes as input a functional decryption key $\mathsf{dk}_{\vec{y}} = (\vec{y}, \vec{d})$, a label ℓ, and ciphertexts. It computes $[\vec{u}_\ell] := \mathcal{H}(\ell)$, $[\alpha] = \sum_i [c_i] \cdot y_i - [\vec{u}_\ell^\top] \cdot \vec{d}$, and eventually solves the discrete logarithm to extract and return α.

5.3.1 Correctness

Note that, as for [7], the result α must be polynomially bounded to efficiently compute the discrete logarithm in the last decryption step: let $\vec{x}, \vec{y} \in \mathbb{Z}_p^n$, we have:

$$\begin{aligned}[\alpha] &= \sum_i [c_i] \cdot y_i - [\vec{u}_\ell^\top] \cdot \vec{d} = \sum_i [\vec{u}_\ell^\top \vec{s}_i + x_i] \cdot y_i - [\vec{u}_\ell^\top] \cdot \sum_i y_i \vec{s}_i \\ &= \sum_i [\vec{u}_\ell^\top] \cdot \vec{s}_i y_i + \sum_i [x_i] \cdot y_i - [\vec{u}_\ell^\top] \cdot \sum_i y_i \vec{s}_i = [\sum_i x_i y_i].\end{aligned}$$

5.3.2 Security Analysis

Theorem 5 (IND-Security) *The above* MCFE *protocol (Section 5.3) is* IND*-secure under the* DDH *assumption,*

$$\mathsf{Adv}^{\mathit{IND}}(\mathcal{A}) \leq 2Q \cdot \mathsf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(t) + \mathsf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(t + 4Q \times t_{\mathbb{G}}) + \frac{2Q}{p},$$

for any adversary $\mathcal{A}$, running within time t, where Q is the number of (direct and indirect —asked by QEncrypt*-queries—) queries to $\mathcal{H}$ (modeled as a random oracle), and $t_{\mathbb{G}}$ is the time for an exponentiation in $\mathbb{G}$.*

We stress that this theorem supports both adaptive encryption queries and adaptive corruptions.

5.4 Labeled MCFE from Public-Key Single-Input FE

In this section, we present the MCFE scheme with labels as constructed in [1].

5.4.1 Construction

The construction by Abdalla et al. [1] resembles the multi-input FE from [4], in which an inner layer of information-theoretic one-time FE is combined with an outer layer of single-input FE. In [1], the authors managed to extend this paradigm to the setting where the encryption additionally takes a label as an additional input. They do so by replacing the one-time pads with pads which are pseudorandom for all used labels ℓ, using techniques similar to those used in [2] to decentralize the generation of functional secret keys.

The detailed construction is given in Fig. 5 and is based on any single-input FE satisfying two simple structural properties: *Two-step decryption* and *linear encryption*. These properties, originally defined in [4] and recalled below (converted to the public-key setting), are satisfied by all known existing single-input FE for inner products (e.g., [3, 7]).

```
$\mathsf{Setup}(1^\lambda, 1^n)$ :
$\mathsf{pp}_{\mathsf{ipfe}} \leftarrow \mathsf{Setup}^\star_{\mathsf{ipfe}}(1^\lambda, 1^n)$, with $L$ implicitly defined from $\mathsf{pp}_{\mathsf{ipfe}}$
Return $\mathsf{pp} = \mathsf{pp}_{\mathsf{ipfe}}$

$\mathsf{KeyGen}(\mathsf{pp})$ :
$(\mathsf{msk}_{\mathsf{ipfe}}, \mathsf{pk}_{\mathsf{ipfe}}) \leftarrow \mathsf{KeyGen}_{\mathsf{ipfe}}(\mathsf{pp}_{\mathsf{ipfe}})$; $\mathsf{msk} := \mathsf{msk}_{\mathsf{ipfe}}$
For $i \in [n]$, $j > i : \mathsf{K}_{i,j} = \mathsf{K}_{j,i} \leftarrow \{0,1\}^\lambda$
Return $\{\mathsf{sk}_i = (\mathsf{pk}, \{\mathsf{K}_{i,j}\}_{j \in [n]})\}_{i \in [n]}$ and $\mathsf{msk}$

$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \vec{x}_i \in \mathcal{R}^m, \ell \in \mathsf{Labels})$ :
Parse $\mathsf{sk}_i = (\mathsf{pk}_{\mathsf{ipfe}}, \{\mathsf{K}_{i,j}\}_{j \in [n]})$
$\vec{t}_{i,\ell} := \sum_{j \neq i} (-1)^{j<i} \mathsf{PRF}_{\mathsf{K}_{i,j}}(\ell) \in \mathbb{Z}_L^{mn}$
$\vec{w}_i := (\mathbf{0} \| \ldots \| \mathbf{0} \| \vec{x}_i \| \mathbf{0} \| \ldots \| \mathbf{0}) + \vec{t}_{i,\ell} \mod L$
$\mathsf{ct}_i \leftarrow \mathsf{Enc}_{\mathsf{ipfe}}(\mathsf{pp}_{\mathsf{ipfe}}, \mathsf{pk}_{\mathsf{ipfe}}, \vec{w}_i)$
Return $\mathsf{ct}_i$

$\mathsf{KeyDer}(\mathsf{pp}, \mathsf{msk}, \vec{y} \in \mathcal{R}^{mn})$ :
Return $\mathsf{sk}_{\vec{y}} \leftarrow \mathsf{KeyDer}_{\mathsf{ipfe}}(\mathsf{pp}_{\mathsf{ipfe}}, \mathsf{msk}_{\mathsf{ipfe}}, \vec{y})$

$\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}_{\vec{y}}, \{\mathsf{ct}_i\}_{i \in [n]})$ :
For $i \in [n]$, $\mathcal{E}(\langle \vec{w}_i, \vec{y} \rangle \mod L, \mathsf{noise}_i) \leftarrow \mathsf{Dec}_{\mathsf{ipfe},1}(\mathsf{pp}_{\mathsf{ipfe}}, \mathsf{sk}_{\vec{y}}, \mathsf{ct}_i)$
Return $\mathsf{Dec}_{\mathsf{ipfe},2}(\mathsf{pp}_{\mathsf{ipfe}}, \mathcal{E}(\langle \vec{w}_1, \vec{y} \rangle \mod L, \mathsf{noise}_1) \circ \cdots \circ \mathcal{E}(\langle \vec{w}_n, \vec{y} \rangle \mod L, \mathsf{noise}_n))$
```

Figure 5: Inner-Product MCFE for $\mathcal{F}_\rho$, where $\rho = (\mathbb{Z}, n, m, X, Y)$, built from a public-key FE := ($\mathsf{Setup}_{\mathsf{ipfe}}$, $\mathsf{Enc}_{\mathsf{ipfe}}$, $\mathsf{KeyDer}_{\mathsf{ipfe}}$, $\mathsf{Dec}_{\mathsf{ipfe}}$) for $\mathcal{F}_{\rho_{\mathsf{ipfe}}}$, where $\rho_{\mathsf{ipfe}} = (\mathbb{Z}, 1, n \cdot m, 2X, Y)$.

Definition 16 (Two-step decryption [4]) *A public-key FE scheme* FE = (Setup, KeyGen, KeyDer, Enc, Dec) *for the function ensemble* $\mathcal{F}_\rho^{\mathsf{ip}}$, *where* $\rho = (\mathbb{Z}, 1, m, X, Y)$, *satisfies the* two-step decryption *property if it admits PPT algorithms* $\mathsf{Setup}^\star$, Dec_1, Dec_2 *and an encoding function* $\mathcal{E}$ *such that:*

1. *For all* $\lambda \in \mathbb{N}$, $\mathsf{Setup}^{\star}(1^{\lambda}, 1^{n})$ *outputs* pp *where* pp *includes* $\rho = (\mathbb{Z}, 1, m, X, Y)$ *and a bound* $B \in \mathbb{R}^{+}$, *as well as the description of a group* $\mathbb{G}$ *(with group law* $\circ$*) of order* $L > n \cdot m \cdot X \cdot Y$, *which defines the encoding function* $\mathcal{E} : \mathbb{Z}_L \times \mathbb{Z} \longrightarrow \mathbb{G}$.

2. *For all* $(\mathsf{msk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$, $\vec{x} \in \mathbb{Z}^m$, $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, \vec{x})$, $\vec{y} \in \mathbb{Z}^m$, *and* $\mathsf{sk} \leftarrow \mathsf{KeyDer}(\mathsf{msk}, \vec{y})$, *we have*
$$\mathsf{Dec}_1(\mathsf{pp}, \mathsf{sk}, \mathsf{ct}) = \mathcal{E}(\langle \vec{x}, \vec{y} \rangle \bmod L, \mathsf{noise}) \ ,$$
for some $\mathsf{noise} \in \mathbb{Z}$ *that depends on* ct *and* sk. *Furthermore, it holds that* $Pr[|\mathsf{noise}| < B] = 1 - \mathsf{negl}(\lambda)$, *where the probability is taken over the random coins of* KeyGen *and* KeyDer. *Note that there is no restriction on the norm of* $\langle \vec{x}, \vec{y} \rangle$ *here.*

3. *The encoding* $\mathcal{E}$ *is linear, that is: for all* $\gamma, \gamma' \in \mathbb{Z}_L$, $\mathsf{noise}, \mathsf{noise}' \in \mathbb{Z}$, *we have*
$$\mathcal{E}(\gamma, \mathsf{noise}) \circ \mathcal{E}(\gamma', \mathsf{noise}') = \mathcal{E}(\gamma + \gamma' \bmod L, \mathsf{noise} + \mathsf{noise}') \ .$$

4. *For all* $\gamma < n \cdot m \cdot X \cdot Y$, *and* $|\mathsf{noise}| < n \cdot B$, $\mathsf{Dec}_2(\mathsf{pp}, \mathcal{E}(\gamma, \mathsf{noise})) = \gamma$.

Definition 17 (Linear encryption [4]) *A secret-key FE scheme* $\mathsf{FE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ *is said to satisfy the linear encryption property if there exists a deterministic algorithm* Add *that takes as input a ciphertext and a message, such that for all* $\vec{x}, \vec{x}' \in \mathbb{Z}^m$, *the following are identically distributed:*
$$\mathsf{Add}(\mathsf{Enc}(\mathsf{pp}, \mathsf{msk}, \vec{x}), \vec{x}'), \quad \textit{and} \quad \mathsf{Enc}\big(\mathsf{pp}, \mathsf{msk}, (\vec{x} + \vec{x}' \bmod L)\big) \ .$$
Recall that the value $L \in \mathbb{N}$ *is defined as part of the output of the algorithm* $\mathsf{Setup}^{\star}$ *(see the two-step decryption property above).*

The proof of correctness for the scheme described in Fig. 5 can be found in [1].

5.4.2 Security Analysis

The security of the new construction is captured by two different theorems in [1]. The first one considers security under static corruption and the other one covers adaptive corruption. We recap these theorems here.

Theorem 6 (sta-pos-IND-security) *If the FE scheme* $\mathsf{FE} = (\mathsf{Setup}_{\mathsf{ipfe}}, \mathsf{KeyGen}_{\mathsf{ipfe}}, \mathsf{KeyDer}_{\mathsf{ipfe}}, \mathsf{Enc}_{\mathsf{ipfe}}, \mathsf{Dec}_{\mathsf{ipfe}})$ *is an* any*-IND-secure FE scheme for the inner product functionality defined as* $\mathcal{F}^{\mathsf{ip}}_{\rho_{\mathsf{ipfe}}}$, *where* $\rho_{\mathsf{ipfe}} = (\mathbb{Z}, 1, m, 2X, Y)$, *and* PRF *is secure, then* MCFE *from Fig. 5 is* sta-pos*-IND-secure for the functionality defined as* $\mathcal{F}^{\mathsf{ip}}_{\rho}$, *where* $\rho = (\mathbb{Z}, n, m, X, Y)$. *Namely, for any PPT adversary* $\mathcal{A}$, *there exist PPT adversaries* $\mathcal{B}$ *and* $\mathcal{B}'$ *such that:*
$$\mathsf{Adv}^{\text{sta-pos-IND}}_{\mathsf{MCFE},\mathcal{A}}(\lambda, n) \leq 2q_{\mathsf{Enc}} \cdot \mathsf{Adv}^{\text{any-IND}}_{\mathsf{FE},\mathcal{B}}(\lambda) + 2(n-1)q_{\mathsf{Enc}} \cdot \mathsf{Adv}_{\mathsf{PRF},\mathcal{B}'}(\lambda),$$
where q_{Enc} *denotes the number of distinct labels queried to* $\mathsf{QLeftRight}$.

Theorem 7 (adt-pos-IND-security) *If the FE scheme* $\mathsf{FE} = (\mathsf{Setup}_{\mathsf{ipfe}}, \mathsf{KeyGen}_{\mathsf{ipfe}}, \mathsf{KeyDer}_{\mathsf{ipfe}}, \mathsf{Enc}_{\mathsf{ipfe}}, \mathsf{Dec}_{\mathsf{ipfe}})$ *is an* any*-IND-secure FE scheme for the inner product functionality defined as* $\mathcal{F}^{\mathsf{ip}}_{\rho_{\mathsf{ipfe}}}$, *where* $\rho_{\mathsf{ipfe}} = (\mathbb{Z}, 1, m, 2X, Y)$, *and* PRF *is secure, then* MCFE *from Fig. 5 is* adt-pos*-IND-secure for the functionality defined as* $\mathcal{F}^{\mathsf{ip}}_{\rho}$, *where* $\rho = (\mathbb{Z}, n, m, X, Y)$. *Namely, for any PPT adversary* $\mathcal{A}$, *there exist PPT adversaries* $\mathcal{B}$ *and* $\mathcal{B}'$ *such that:*
$$\mathsf{Adv}^{\text{adt-pos-IND}}_{\mathsf{MCFE},\mathcal{A}}(\lambda, n) \leq 2(n+1)n(n-1)^2 q_{\mathsf{Enc}} \cdot \mathsf{Adv}_{\mathsf{PRF},\mathcal{B}}(\lambda) + 2(n+1)q_{\mathsf{Enc}} \cdot \mathsf{Adv}^{\text{any-IND}}_{\mathsf{FE},\mathcal{B}'}(\lambda) \ ,$$
where q_{Enc} *denotes the number of distinct labels queried to* $\mathsf{QLeftRight}$.

The proof for both of the theorems can be found in [1].

Using the generic transformation presented in [1], it is possible to remove the pos-security restriction, and obtain adt-any-IND security.

6 Video Surveillance Scenario

Recall that in this use case, our goal is to solve a Local Decision Making (LDM) problem. In the D4.1 we presented an Inner-Product Predicate Encryption (IPPE) scheme for this purpose and a generic transformation, by Katz, Sahai, and Waters in [20], enabling IPPE to compute exact thresholds over the encrypted data. During the second year we realized that hiding the threshold was not a primary concern and so in the deliverable D4.2 we presented two other schemes. The first one was a single-input inner-product functional encryption scheme [7], which requires the camera to only encrypt positive values when creating the ciphertext for the gateway. The second scheme [8] was a FE scheme allowing the computation of quadratic functions, which does not impose any restriction on the values being encrypted by the camera.

In comparison to the schemes proposed in the D4.1 and D4.2, we came to realize that performance was the main issue, and no functional encryption scheme currently would fulfill them. Specifically, the time requirement that we had to achieve was that the gateway must take a decision in under half a second. The sheer amount of motion vectors in videos makes it unrealistic for the current state-of-the-art functional encryption. However, new developments lead us to believe that greatly reducing the number of motion vectors still allows motion detection. Indeed, we now only take into account a few vectors, the ones with the biggest norm and encrypt them with FE. Of course this lessens the precision of motion detection, but it allows for practical performance. With that in mind, schemes based on DDH, which are usually limited by the size of the message to decrypt, become attractive again as DDH-based solutions are in general faster than lattice-based ones. Also, as the camera has to select which vectors to encrypt with FE, a quadratic scheme will usually be slower than an IP scheme. This leaves us with the following DDH-based IP scheme as our best candidate for the video surveillance use case.

6.1 MDDH-based Single-Input Inner-Product Functional Encryption Instantiations

In this section we present a single-input inner-product functional encryption scheme based on the MDDH assumption. Fig. 6 recalls the single-input inner-product FE from [7, Section 3], generalized to the $\mathcal{D}_k$-MDDH setting, as in [5, Figure 15].

```
Setup($1^\lambda, \mathcal{F}_1^{m,X,Y}, \ell$):
$\mathbb{G} := (\mathbb{G}, p, g) \leftarrow \mathbb{G}(1^\lambda)$, $\mathbf{A} \xleftarrow{R} \mathsf{D}_k$, $\mathbf{W} \xleftarrow{R} \mathbb{Z}_p^{m\times(k+1)}$
pk $:= (\mathbb{G}, [\mathbf{A}], [\mathbf{WA}])$, msk $:= \mathbf{W}$
Return (pk, msk)

Enc(pk, $\mathbf{x} \in \mathbb{Z}_p^m$):
$\mathbf{r} \xleftarrow{R} \mathbb{Z}_p^k$, $\mathbf{c} := \begin{pmatrix} -\mathbf{Ar} \\ \mathbf{x} + \mathbf{WAr} \end{pmatrix}$
Return ct$_\mathbf{x} := [\mathbf{c}] \in \mathbb{G}^{k+m+1}$

KeyGen(msk, $\mathbf{y} \in \mathbb{Z}_p^m$):
Return sk$_\mathbf{y} := \begin{pmatrix} \mathbf{W}^\top \mathbf{y} \\ \mathbf{y} \end{pmatrix} \in \mathbb{Z}_p^{k+m+1}$

Dec(pk, ct$_\mathbf{x} := [\mathbf{c}]$, sk$_\mathbf{y}$):
$C := [\mathbf{c}^\top \mathsf{sk}_\mathbf{y}]$
Return $\log(C)$
```

Figure 6: Functional encryption scheme for the class $\mathcal{F}_1^{m,X,Y}$, based on the D_k-MDDH assumption [7, 5].

7 Relation to D3.1 Requirements, D4.1 and D4.2

Deliverable D3.1 outlined the requirements for all the FENTEC use cases being developed in WP7. In response, we presented several schemes that fit the requirements for these use cases in the deliverable D4.1. This document offers new improved schemes, presented in Sections 5 and 6. In this section, we summarize the list of requirements that are met by these schemes.

7.1 Digital Currency Requirements

The CP-ABE scheme presented in Section 4.2 fulfills the requirements **NF-Data Protection-DC.05** and **NF-Data Protection-DC.06**.

NF-Data Protection-DC.05: This *auditability* requirement states that transactions should be auditable with respect to GDPR and with customer privacy in mind. The use of CP-ABE schemes proposed in Section 4.2 meets this goal by allowing customers to specify their own decryption policies.

NF-Data Protection-DC.06: This *auditability agreement* requirement states that customers and merchants must sign a contract which specifies the types of audits and auditors to which they can be subject. This requirement is satisfied because the CP-ABE scheme proposed in Section 4.2 allows for both the customer and the merchant to agree on the policy during the encryption phase since the policy itself is provided in the clear by the encryptor.

In addition to the above, the specification of a KP-ABE scheme given in Section 4.3 is necessary for the fulfillment of requirement **F-Design-DC.11**, which states that the API should implement secure CP and KP ABE schemes.

7.2 Web Analytics Requirements

The DMCFE scheme proposed in D4.1, for the Web Analytics use case, already fulfilled all the requirements it could. The new scheme improves on the security and fill in the same three requirements **NF-Security-WA.01**, **NF-Data Protection-WA.05** and **NF-Performance- WA.06**:

NF-Security- WA.01: This requirement called for a DMCFE scheme with no central authority. In the scheme presented in section Section 5, no central authority is required as both the setup and the key generation algorithms are decentralized and computed interactively by the participants.

NF-Data Protection-WA.05: This requirement is fulfilled due to the fact that the participants generate their own encryption keys and are therefore the only ones who can send encrypted data to the collection point

NF-Performance- WA.06: This requirement asked for alternative solutions and with these new schemes, we now have two decentralized solutions as well as three centralized solutions based on multiple hardness assumptions, some achieving post-quantum security.

7.3 Video Surveillance Requirements

The IPPE scheme proposed in D4.1, for the video surveillance use case, fulfilled four major requirements although we feared the scheme would be impractical. The scheme proposed in Section 6 also fulfills the same four requirements and will improve on the performance of the API. Let us recall the four requirements: **NF-Security- IoT.01**, **NF-Data Protection-IoT.04**, **F-Implementation-IoT.10**, and **NF-Security- IoT.02**.

NF-Security- IoT.01: The *end-to-end encryption* requirement is satisfied because the camera encrypts all the streaming data being generated using the public key of the scheme, whose master secret key is only known to the security center.

NF-Data Protection-IoT.04: The *leakage resilience* requirement is satisfied by both schemes because the functional decryption key given to the gateway only allows the latter to know the sum or quadratic sum of the motion vectors length.

F-Implementation-IoT.10: This requirement is fulfilled by both schemes since the threshold value to which the data will be compared during decryption does not need to be known by the encryptor.

NF-Security- IoT.02: As shown in Section 6, both schemes meet the *IND-CPA security* requirement.

8 Conclusion

In this document, we described the final specifications of functional encryption for the three use cases under consideration in the FENTEC project. These schemes were chosen based on the requirements in Deliverable D3.1.

In the case of the digital currency use case, we specified two attribute-based encryption schemes from [6], depending on whether the access policy is specified during encryption or key generation. For the web analytics use case, we proposed two kinds of DMCFE schemes able to compute inner products, with or without encryption labels. The DMCFE schemes proposed in both cases can be instantiated under the DDH assumption and under quantum-safe lattice assumptions. This fulfills all the requirements elicited in D3.1 of the project. Finally, in the video surveillance use case, we proposed a trade-off between efficiency and the precision of motion vectors that greatly reduces the data needed to detect movement and decreases the overhead incurred by the use of functional encryption schemes. While all but the performance requirements had already been met, this new approach means that the previously considered scheme also achieves the performance requirements.

References

[1] Michel Abdalla, Fabrice Benhamouda, and Romain Gay. From single-input to multi-client inner-product functional encryption. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 552–582, Kobe, Japan, December 8–12, 2019. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-34618-8_19. (Pages v, 16, 19, 20, 21, and 22.)

[2] Michel Abdalla, Fabrice Benhamouda, Markulf Kohlweiss, and Hendrik Waldner. Decentralizing inner-product functional encryption. In Dongdai Lin and Kazue Sako, editors, *PKC 2019: 22nd International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 11443 of *Lecture Notes in Computer Science*, pages 128–157, Beijing, China, April 14–17, 2019. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-17259-6_5. (Pages 16, 17, 18, and 20.)

[3] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In Jonathan Katz, editor, *PKC 2015: 18th International Conference on Theory and Practice of Public Key Cryptography*, volume 9020 of *Lecture Notes in Computer Science*, pages 733–751, Gaithersburg, MD, USA, March 30 – April 1, 2015. Springer, Heidelberg, Germany. doi:10.1007/978-3-662-46447-2_33. (Page 20.)

[4] Michel Abdalla, Dario Catalano, Dario Fiore, Romain Gay, and Bogdan Ursu. Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 597–627, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-96884-1_20. (Pages 20 and 21.)

[5] Michel Abdalla, Romain Gay, Mariana Raykova, and Hoeteck Wee. Multi-input inner-product functional encryption from pairings. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 601–626, Paris, France, April 30 – May 4, 2017. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-56620-7_21. (Pages iii, 23, and 24.)

[6] Shashank Agrawal and Melissa Chase. FAME: Fast attribute-based message encryption. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 665–682, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press. doi:10.1145/3133956.3134014. (Pages v, 1, 12, 13, 14, and 26.)

[7] Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 333–362, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany. doi:10.1007/978-3-662-53015-3_12. (Pages iii, 1, 19, 20, 23, and 24.)

[8] Carmen Elisabetta Zaira Baltico, Dario Catalano, Dario Fiore, and Romain Gay. Practical functional encryption for quadratic functions with applications to predicate encryption. In Jonathan Katz and

Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 67–98, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany. doi:10.1007/978-3-319-63688-7_3. (Pages 1 and 23.)

[9] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In Bart Preneel and Stafford Tavares, editors, *SAC 2005: 12th Annual International Workshop on Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331, Kingston, Ontario, Canada, August 11–12, 2006. Springer, Heidelberg, Germany. doi:10.1007/11693383_22. (Page 4.)

[10] Mihir Bellare, Anand Desai, Eric Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science*, pages 394–403, Miami Beach, Florida, October 19–22, 1997. IEEE Computer Society Press. doi:10.1109/SFCS.1997.646128. (Page 10.)

[11] Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart. *Advances in Elliptic Curve Cryptography*. Cambridge University Press, 2005. (Page 4.)

[12] Jérémy Chotard, Edouard Dufour Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. Decentralized multi-client functional encryption for inner product. Cryptology ePrint Archive, Report 2017/989, 2017. http://eprint.iacr.org/2017/989. (Page 19.)

[13] Jérémy Chotard, Edouard Dufour Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. Decentralized multi-client functional encryption for inner product. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 703–732, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-03329-3_24. (Pages 9 and 10.)

[14] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 129–147, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-40084-1_8. (Page 3.)

[15] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23(2):224–280, April 2010. doi:10.1007/s00145-009-9048-z. (Page 4.)

[16] S.D. Galbraith, K.G. Paterson, and N.P. Smart. Pairings for cryptographers. Cryptology ePrint Archive, Report 2006/165, 2006. http://eprint.iacr.org/2006/165. (Page 13.)

[17] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 578–602, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-55220-5_32. (Page 7.)

[18] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006: 13th Conference on Computer and Communications Security*, pages 89–98, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press. Available as Cryptology ePrint Archive Report 2006/309. doi:10.1145/1180405.1180418. (Pages 6, 7, and 12.)

[19] Aurore Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In Michael J. Jacobson Jr., Michael E. Locasto, Payman Mohassel, and Reihaneh Safavi-Naini, editors, *ACNS 13: 11th International Conference on Applied Cryptography and Network Security*, volume 7954 of *Lecture Notes in Computer Science*, pages 357–372, Banff, AB, Canada, June 25–28, 2013. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-38980-1_22. (Page 4.)

[20] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. Cryptology ePrint Archive, Report 2007/404, 2007. http://eprint.iacr.org/2007/404. (Page 23.)

[21] NIST. Recommendation for key management - part 1: General sp 800-57. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf, january 2016. (Page 3.)

[22] ECRYPT2 MAYA Project. Final report on main computational assumptions in cryptography. http://www.ecrypt.eu.org/ecrypt2/documents/D.MAYA.6.pdf, January 2013. (Page 4.)

[23] Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany. doi:10.1007/11426639_27. (Page 6.)