cosmian

Covercrypt

A Traceable Attribute-Based Encryption with PQ/T Hybrid Security

David Pointcheval (Cosmian)
Joint work with Théophile Brézot, Chloé Hébant and Paola de Perthuis

Data Centric Security

Public-Key Encryption (PKE)

With PKE, a data owner can encrypt for a **specific** target recipient user, from his ID/Public Key
But one may want to target **groups** of users, according to their roles/activities/status/rights.

Attribute-Based Encryption (ABE)

ABE has been proposed for this general task:

users (their keys) and **data** (their encryptions) are associated to attributes **Y** and policies **P**

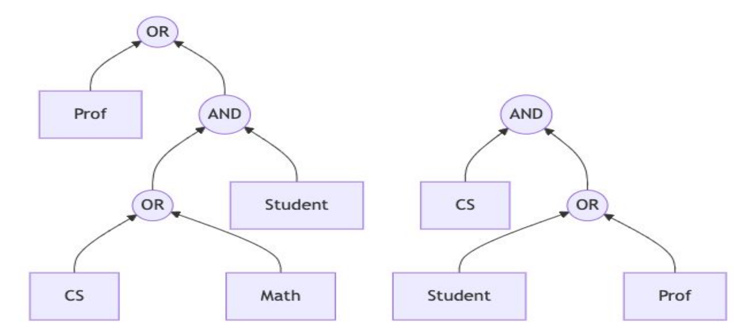
- Attributes **Y** = $\{b_1, \ldots, b_n\}$, a statement of Boolean variables (true or false)
- Policy **P** = a Boolean formula on these variables

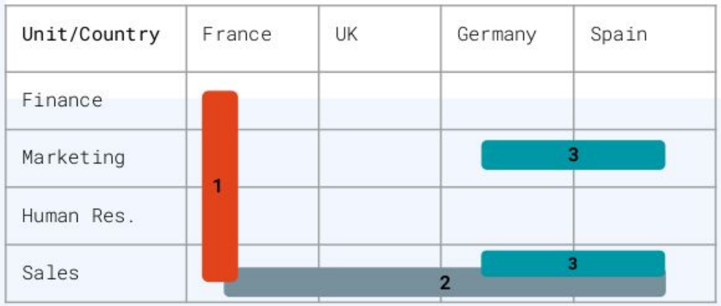
if **P**(**Y**) is true, the user can access the data (i.e. the key can decrypt the ciphertext)

Two kinds of ABE have been defined:

- **Key-Policy ABE**: the key depends on the policy, and the ciphertext is associated to attributes
- **Ciphertext-Policy ABE**: the key is associated to attributes, and the ciphertext depends on the policy

Access-Structures





Complex Access Structures

- **Y** = {Prof, Physics} : OK
- **Y** = {Student, Physics} : KO

P could even be non-monotonous (with NOT-gates)

- [GPSW06]: KP-ABE
- [BSW07]: CP-ABE

Require either

- Pairing-friendly curves for ECDH
- Huge ciphertexts with LWE

Real-World Access Structures

- Multi-dimensional structure
- Independent or hierarchical values
- Right = point in the space (e.g. `UK-Sales`)
- Attribute = set of points/rights (e.g. 1)

Main objectives

- Encompass most of the natural use-cases
- Post-quantum transition
- Compact ciphertexts
- Crypto agility

Covercrypt

- Use of optimal subset-cover of the rights for short ciphertexts
- KEM/DEM hybrid approach for efficiency
  - KEM to be combined with any Authenticated Encryption as DEM
- PQ/T hybrid KEM for security (PQ migration)
- Construction with any KEMs in black-box

An AB-KEM from any KEM

A Key Encapsulation Mechanism

Formalism

- KEM.KeyGen(1^k): **key generation** with output (pk,sk)
- KEM.Enc(pk): **encapsulation** with output (E,K)
- KEM.Dec(sk,E): **decapsulation** of E with output K

Security

- SK-IND = Session-Key Indistinguishability: $(E,K) \approx (E,\$)$ — Privacy of the session key
- PK-IND = Public-Key Indistinguishability: $(E,pk) \approx (E,pk')$ — Anonymity

Attributes and AB-KEM

For any attribute **a** in the universe **A**: $(pk_\mathbf{a},sk_\mathbf{a}) \leftarrow \text{KEM.KeyGen}(1^k)$

For any user U with attributes $\mathbf{Y} \subset \mathbf{A}$: $SK_U = \{\, sk_\mathbf{a},\ \mathbf{a} \in \mathbf{Y} \,\}$

For an **encapsulation** under attributes $\mathbf{X} \subset \mathbf{A}$: $E = \{\, (E_\mathbf{a}, F_\mathbf{a}),\ \mathbf{a} \in \mathbf{X} \,\}$
with $K \leftarrow \mathbf{K}$, and $(E_\mathbf{a}, K_\mathbf{a}) \leftarrow \text{KEM.Enc}(pk_\mathbf{a})$, $F_\mathbf{a} \leftarrow K \oplus K_\mathbf{a}$

For **decapsulation** under a key for $\mathbf{Y} \subset \mathbf{A}$: if $\mathbf{X} \cap \mathbf{Y} \neq \emptyset$, there is at least a common **a** in $\mathbf{X} \cap \mathbf{Y}$
$K_\mathbf{a} \leftarrow \text{KEM.Dec}(sk_\mathbf{a},E_\mathbf{a})$, and $K \leftarrow F_\mathbf{a} \oplus K_\mathbf{a}$

An AB-KEM from any KEM: Security and Efficiency

AB-KEM

- For an **encapsulation** under attributes $\mathbf{X} \subset \mathbf{A}$: $E = \{ (E_\mathbf{a}, F_\mathbf{a}),\ \mathbf{a} \in \mathbf{X} \}$
  with $K \leftarrow \mathbf{K}$, and $(E_\mathbf{a}, K_\mathbf{a}) \leftarrow$ KEM.Enc($pk_\mathbf{a}$), $F_\mathbf{a} \leftarrow K \oplus K_\mathbf{a}$, for all $\mathbf{a} \in \mathbf{X}$
- For **decapsulation** under a key for $\mathbf{Y} \subset \mathbf{A}$: if there is at least a common **a** in $\mathbf{X} \cap \mathbf{Y}$
  $K_\mathbf{a} \leftarrow$ KEM.Dec($sk_\mathbf{a}, E_\mathbf{a}$), and $K \leftarrow F_\mathbf{a} \oplus K_\mathbf{a}$

Security

- SK-IND-CPA KEM ⇒ SK-IND-CPA AB-KEM: Session-key privacy
- PK-IND-CPA KEM ⇒ AC-IND-CPA AB-KEM: Access-control privacy

How to find the good **a**?

CCA Security?

- For an **encapsulation** under $\mathbf{X} \subset \mathbf{A}$: $E = \{ V, (E_\mathbf{a}, F_\mathbf{a}),\ \mathbf{a} \in \mathbf{X} \}$
  with S random, and $(E_\mathbf{a}, K_\mathbf{a}) \leftarrow$ KEM.Enc($pk_\mathbf{a}$), $F_\mathbf{a} \leftarrow S \oplus \mathcal{H}(K_\mathbf{a}, \{E_\mathbf{a}\}_\mathbf{a})$, for all $\mathbf{a} \in \mathbf{X}$, then $(K, V) \leftarrow \mathcal{H}'(S, \{ (E_\mathbf{a}, F_\mathbf{a}) \}_\mathbf{a})$
- For **decapsulation** under a key for $\mathbf{Y} \subset \mathbf{A}$, check for all the possible pairs ($\mathbf{b} \in \mathbf{Y}$, **a**)
  $K'' \leftarrow$ KEM.Dec($sk_\mathbf{b}, E_\mathbf{a}$), and $S'' \leftarrow F_\mathbf{a} \oplus \mathcal{H}(K'', \{E_\mathbf{a}\}_\mathbf{a})$, then $(K', V') \leftarrow \mathcal{H}'(S'', \{ (E_\mathbf{c}, F_\mathbf{c}) \}_\mathbf{c})$: if $V' = V$ we have $K' = K$

CCA KEM ⇒ CCA AB-KEM and **efficient decryption**

PQ/T Hybridization of KEMs

Hybrid KEM: CCA-KEM & CPA-KEM'

- **HKEM**.KeyGen(1^k): run $(pk,sk) \leftarrow$ KEM.KeyGen(1^k)
  and $(pk',sk') \leftarrow$ KEM'.KeyGen(1^k)

- **HKEM**.Enc(pk,pk'): with *S random*, $(E,K) \leftarrow$ KEM.Enc(pk),
  and $(E',K') \leftarrow$ KEM'.Enc($pk,\mathcal{G}(S)$),
  then $F \leftarrow S \oplus \mathcal{F}(K,K',E',E)$

  Encapsulation = (E',E,F) for the key $K'' = \mathcal{H}(S,E',E,F)$

- **HKEM**.Dec($(sk,sk'),(E',E,F)$): $K \leftarrow$ KEM.Dec(sk,E), $K' \leftarrow$ KEM'.Dec(sk',E'),
  and $S \leftarrow F \oplus \mathcal{F}(K,K',E',E)$, with output key $K'' = \mathcal{H}(S,E',E,F)$
  if and only if (E',K')= KEM'.Enc($pk',\mathcal{G}(S)$)

- KEM = ML-KEM (D-MLWE)
- KEM' = Hash-ECDH (EC-CDH)

Security

- KEM is CCA
- KEM' + FO transform is CCA : from any Non-Interactive Key Exchange (NIKE)
- **Combination is CCA**

CCA Hybrid Security

Additional Features

Traceability

Data Centric Security: Keys are only specific to attributes/rights, but not to users

In the Diffie-Hellman (NIKE) part, one can efficiently specialize keys for users, to identify abuses

On can then use the **Boneh-Franklin traceability** mechanism (Crypto '99) to deal with coalitions of traitors

- white-box traceability (find a least one traitor from the extracted key)
- black-box confirmation (confirm a candidate list of traitors just interacting with the pirate box)

Tool for Policy Conversion

For a **Ciphertext-Policy** AB-KEM, one needs to convert a policy into a **small** set **X**: efficient tool

```
Policy: {
  Security: {
    None,
    Medium,
    High
  },
  Country: {
    France,
    Germany,
    UK,
    Spain
  }
}
```

According to the expected meaningful target sets of right, one will generate multiple attributes, to reduce the number of attributes at encryption time: **smaller encapsulations, short encapsulation and decapsulation times**

Benchmarks

Encapsulation Size

- ECDH on R25519 + MLKEM512 : **16.(2*t* + 3) + 800.|X|** bytes
- ECDH on R25519 + MLKEM768 : **16.(2*t* + 3) + 1120.|X|** bytes

where *t* is the tracing threshold and **X** is the target set of rights: **X** aimed to be small

Timings (on an M1 CPU):

- ECDH on R25519 + MLKEM512
  - Encapsulation : **~100µs** for |**X**| = 1 (+45µs per additional attribute)
  - Decapsulation : **~250µs** for |**Y**| = 12 (and |**X**| = 1), on average
- ECDH on R25519 + MLKEM768
  - Encapsulation : **~110µs** for |**X**| = 1 (+55µs per additional attribute)
  - Decapsulation : **~260µs** for |**Y**| = 12 (and |**X**| = 1), on average

Conclusion

Covercrypt

- Efficient AB-KEM with PQ/T hybridization and CCA security
    - with any KEMs in black-box
    - conversion from policies to sets of rights
- Traceability
- Full key life-cycle management
    - when removing some user rights
    - when adding some dimensions or attributes in the system
    - etc

- Approved by ETSI as a Standard: to be published soon
    - with ECDH and MLKEM
- Formal security analysis: available soon on ePrint
- Implementation available on Public Cosmian GitHub
    - quite efficient with Ristretto25519 and ML-KEM512/768/1024

https://github.com/Cosmian
https://docs.cosmian.com/