From theoretical crypto to practice: gloups an abominable gap

Cryptie, Oblazy

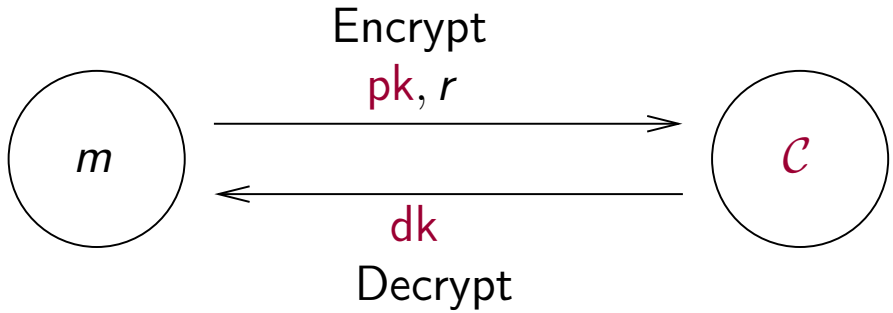
1. Encryption and Signature: Just a 2 min reminder
2. Libraries
3. Funny Cryptography

1. Encryption and Signature: Just a 2 min reminder
2. Libraries
3. Funny Cryptography

Definition (Encryption Scheme)

$\mathcal{E} = (\text{Setup}, \text{EKeyGen}, \text{Encrypt}, \text{Decrypt})$:

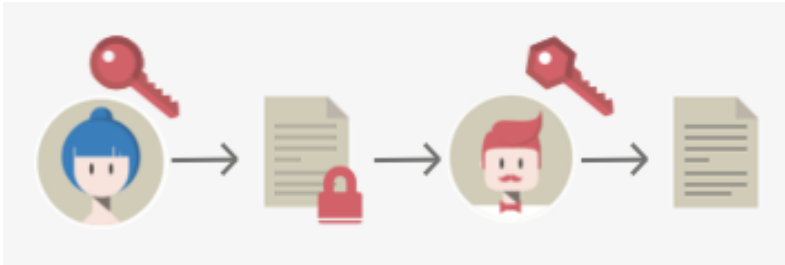
- $\text{Setup}(1^{\lambda})$: param;
- EKeyGen(param): public *encryption* key pk, private *decryption* key dk;
- Encrypt(pk, m; r): ciphertext c on $m \in \mathcal{M}$ and pk;
- Decrypt(dk, c): decrypts c under dk.

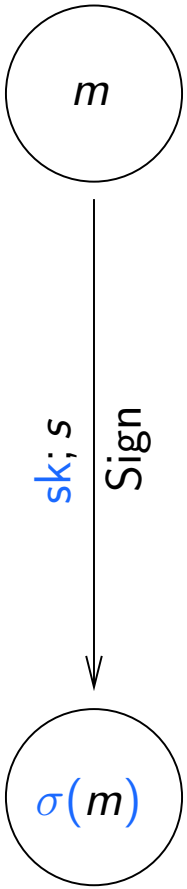


Indistinguishability:
Given M_0, M_1, it should be hard to guess which one is encrypted in C.

Definition

An assymetric encryption scheme allows Cryptie, using the public key of Bob, to encrypt a message to Bob in such a way that only Bob, with his secret key, can read it.





Definition (Signature Scheme)

$\mathcal{S} = (\text{Setup}, \text{SKeyGen}, \text{Sign}, \text{Verif})$:

- $\text{Setup}(1^{\mathfrak{K}})$: param;
- SKeyGen(param): public *verification* key vk, private *signing* key sk;
- Sign(sk, m; s): signature σ on m, under sk;
- Verif(vk, m, σ): checks whether σ is valid on m.

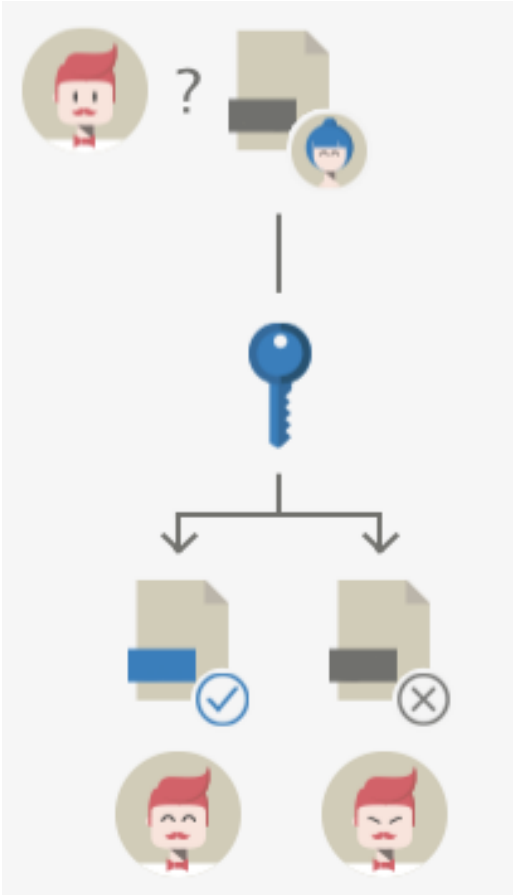
Unforgeability:
Given q pairs (m_i, σ_i), it should be hard to output a valid σ on a fresh m.

Definition

A signature scheme allows Cryptie, using her secret key, to sign a document in such a way that anybody knowing her public key, for example Bob, can be sure that she signs exactly this document.

Definition

A signature scheme allows Cryptie, using her secret key, to sign a document in such a way that anybody knowing her public key, for example Bob, can be sure that she signs exactly this document.



Libre Crypto libraries? we have a lot of them

NaCL	Public domain
Botan	(simplified) BSD
Boncycastle	MIT License
Cryptlib	Sleepycat License
Crypto++	Boost Software License 1.0 (Public domain for files)
Libgcrypt	LGPLv2.1+
Libtomcrypt	Public License and WTFPL
Nettle	GPLv2+ and LGPLv3+
OpenSSL and LibreSSL	OpenSSL License, original SSLeay Licence
etc ...	

$\Rightarrow$ You can even discover some new Free Software license !

$\Rightarrow$ Mostly vanilla crypto...

$\Rightarrow$ Community knows the good parameter, the good curve but...

Academical crypto in real world

When academics says "this is broken", it is patched (nearly in a timely manner).

Academical crypto in real world

When academics says "this is broken", it is patched (nearly in a timely manner).

Example

- First theoretical academic attack on SHA-1 in 2005
- First academic attack that may(?) be used 2010-2015ish.
- Start of the end of SHA-1 2013-2015.
- Summer 2016: *Practical* attacks.

Academical crypto in real world 2

What about funny crypto?

20+ years later the lucky ones are just starting to be used (in weird Blockchains).

What kind of strange properties can we have?

- Weird signatures
- Strange encryption
- Crazy stuff

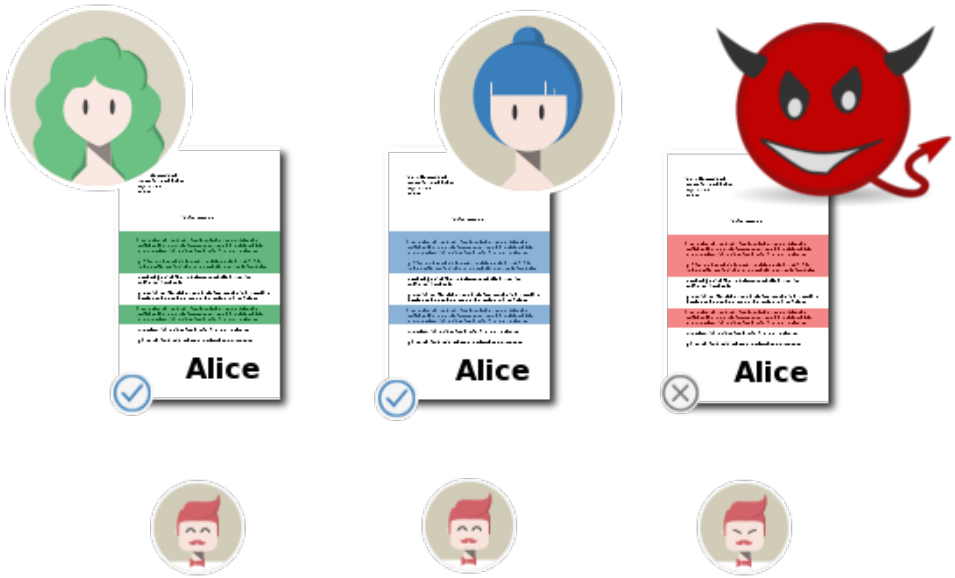
$\Rightarrow$ Let's talk about funny crypto

Weird signatures

Sanitizable Signatures [KR00]

Definition

A sanitizable signature allows Alice to signs a text in such a way that she can give Cryptie the right to modify some parts of it while keeping a correct signature of her on this modified message.



Group Signatures [CvH91]

Definition

A group signature allows Bob to signs as a member of a group in such a way that only a special (optional) entity, an "Opener", would be able to know that HE was the signer of the given message.

Group Signatures [CvH91]

Definition

A group signature allows Bob to signs as a member of a group in such a way that only a special (optional) entity, an "Opener", would be able to know that HE was the signer of the given message.

Group Signatures [CvH91]

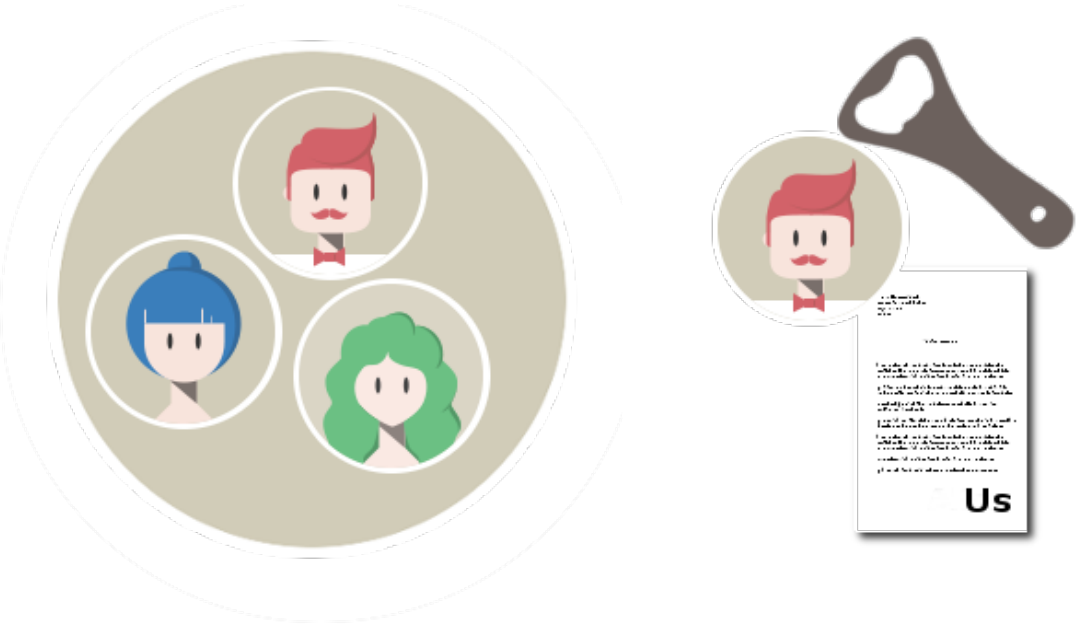
Definition

A group signature allows Bob to signs as a member of a group in such a way that only a special (optional) entity, an "Opener", would be able to know that HE was the signer of the given message.

Group Signatures [CvH91]

Definition

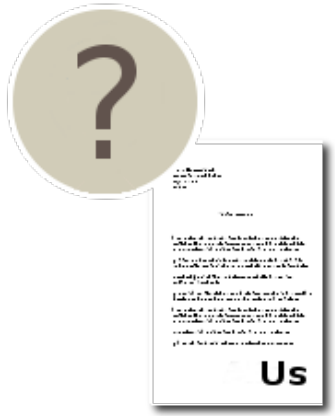
A group signature allows Bob to signs as a member of a group in such a way that only a special (optional) entity, an "Opener", would be able to know that HE was the signer of the given message.

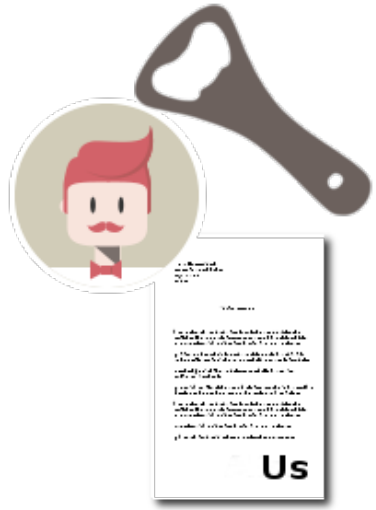


~~Group~~ *Ring* Signatures [RST01]

Definition

A ~~group~~ *ring* signature allows Bob to signs as a member of a group, *that he built alone*, in such a way that ~~only a special (optional) entity, an "Opener",~~ *no one* would be able to know that HE was the signer of the given message.





~~Group~~ *Ring* Signatures [RST01]

Definition

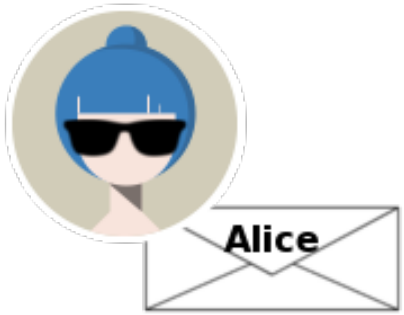
A ~~group~~ *ring* signature allows Bob to signs as a member of a group, *that he built alone*, in such a way that ~~only a special (optional) entity, an "Opener",~~ *no one* would be able to know that HE was the signer of the given message.

The only technology using it is some Blockchain implementation...

Blind Signatures [Chaum83]

Definition

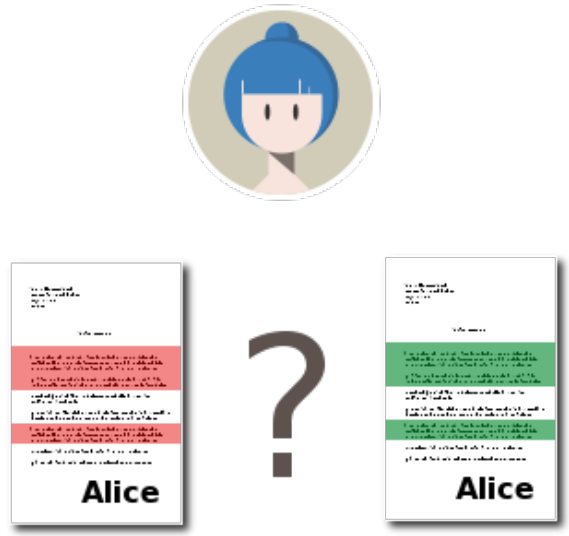
A blind signature allows Alice to signs a letter "through" its envelope. If later, she sees two documents she signs, she won't be able to know which text she signs when.



Blind Signatures [Chaum83]

Definition

A blind signature allows Alice to signs a letter "through" its envelope. If later, she sees two documents she signs, she won't be able to know which text she signs when.



Strange encryption

Homomorphic Encryption [RSA77]

Definition

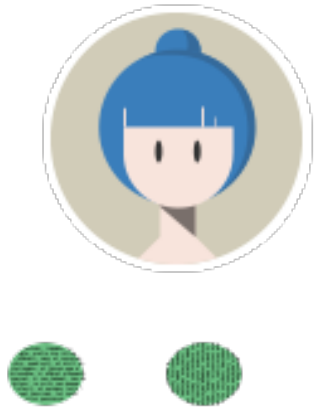
In an Homomorphic Encryption, a user encrypts a message M, using a public encryption key. The resulting ciphertext can then be decrypted using a secret decryption key. *Ciphertexts can be combined, so that the decryption leads to the combination of the plaintext*

42 4

Homomorphic Encryption [RSA77]

Definition

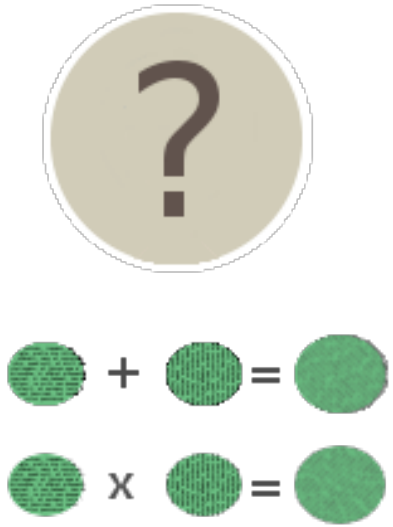
In an Homomorphic Encryption, a user encrypts a message M, using a public encryption key. The resulting ciphertext can then be decrypted using a secret decryption key. *Ciphertexts can be combined, so that the decryption leads to the combination of the plaintext*



Homomorphic Encryption [RSA77]

Definition

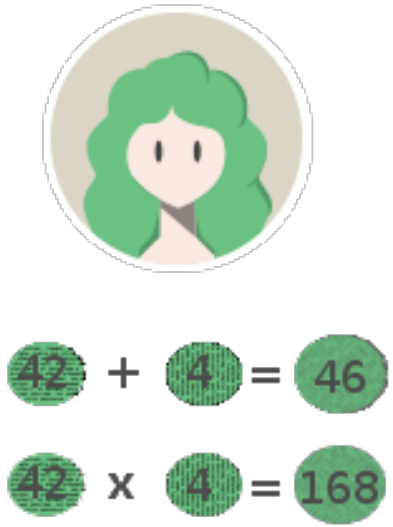
In an Homomorphic Encryption, a user encrypts a message M, using a public encryption key. The resulting ciphertext can then be decrypted using a secret decryption key. *Ciphertexts can be combined, so that the decryption leads to the combination of the plaintext*



Homomorphic Encryption [RSA77]

Definition

In an Homomorphic Encryption, a user encrypts a message M, using a public encryption key. The resulting ciphertext can then be decrypted using a secret decryption key. *Ciphertexts can be combined, so that the decryption leads to the combination of the plaintext*



Threshold Encryption [DDFY94]

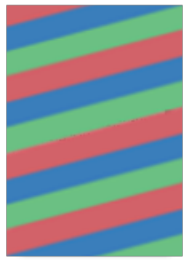
Definition

In a Threshold Encryption, a user encrypts a message M, using a public encryption key. The resulting ciphertext can then be decrypted using **at least k** secret decryption keys.

Threshold Encryption [DDFY94]

Definition

In a Threshold Encryption, a user encrypts a message M, using a public encryption key. The resulting ciphertext can then be decrypted using **at least k** secret decryption keys.



Threshold Encryption [DDFY94]

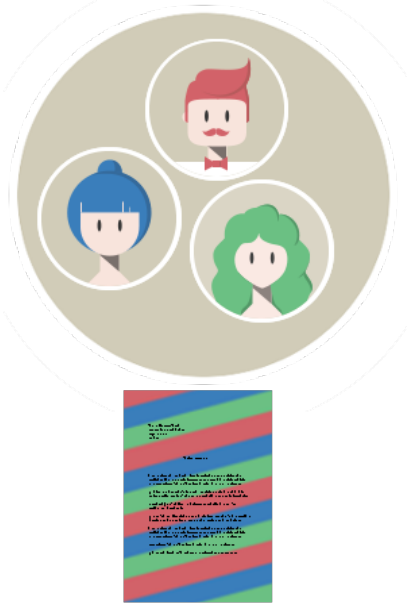
Definition

In a Threshold Encryption, a user encrypts a message M, using a public encryption key. The resulting ciphertext can then be decrypted using **at least k** secret decryption keys.

Threshold Encryption [DDFY94]

Definition

In a Threshold Encryption, a user encrypts a message M, using a public encryption key. The resulting ciphertext can then be decrypted using **at least k** secret decryption keys.



Broadcast Encryption [FN94]

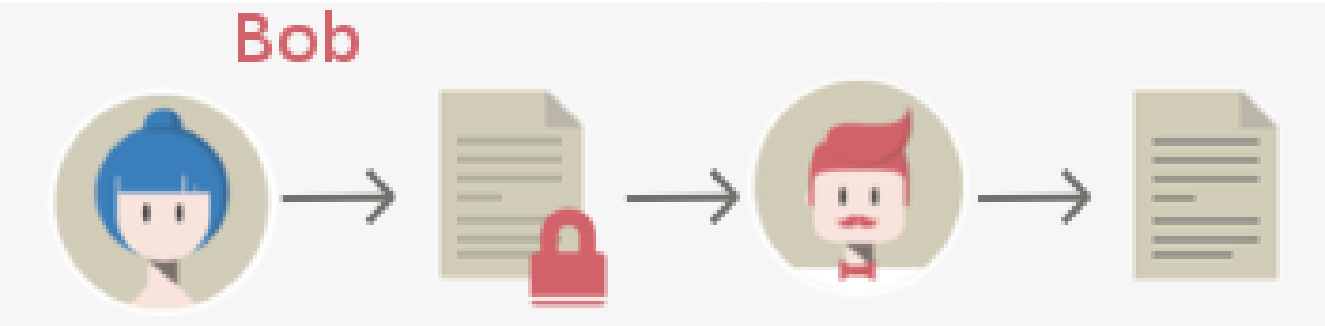
Definition

In a Broadcast Encryption, a user encrypts a message M for a subset of users. The resulting ciphertext can then be decrypted using one of k secret decryption key.

Identity-based Encryption [Sha01]

Definition

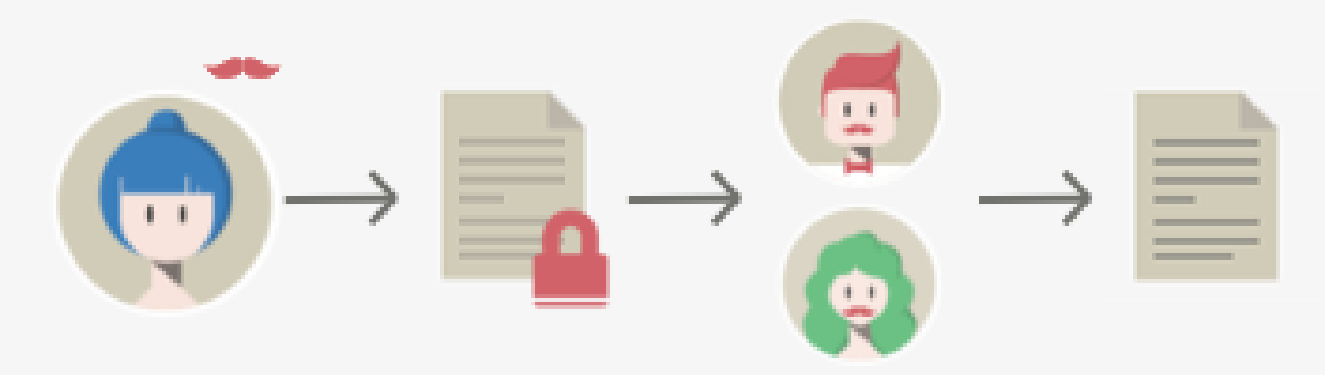
In an Identity-based Encryption, a user encrypts a message M, using a ~~public encryption key~~ user identity. The resulting ciphertext can then be decrypted using a secret decryption key.



Attribute-base Encryption [SW04]

Definition

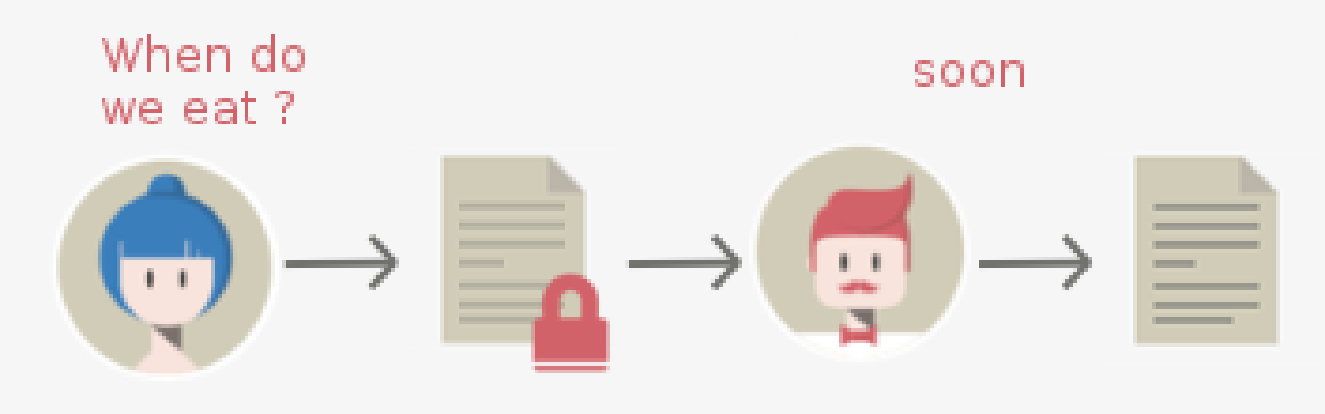
In an Attribute-based Encryption, a user encrypts a message M, using a public encryption key corresponding to some policy. The resulting ciphertext can then be decrypted using a ~~secret decryption key~~ credential fitting the policy.



Witness Encryption [GGSW13]

Definition

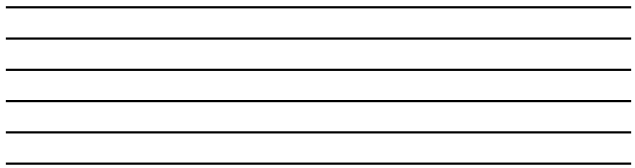
In a Witness Encryption, a user encrypts a message M, using a public encryption key. The resulting ciphertext can then be decrypted using a ~~secret decryption key~~ witness of some property.



Crazy stuff


ZKPK
Key
Language
SPHF
Signature
Envelope
Transfer
Credential
Hash
Authenticated
Exchange
Solution
Oblivious
Handshake
Password
NIWI
Interval
SNARK
Secret

Zero-Knowledge Proof [GMR85]



Alice

Bob

Interactive method for Alice to prove to Bob that she knows something $\mathcal{S}$ without revealing anything other than this fact.

Definition

Functions that can be evaluated in two different ways, either with a *secret* hashing key hk or with a *public* projected key hp and a secret witness

$K = \mathrm{Hash}_L(\mathrm{hk}; x)$

Word x

Language L

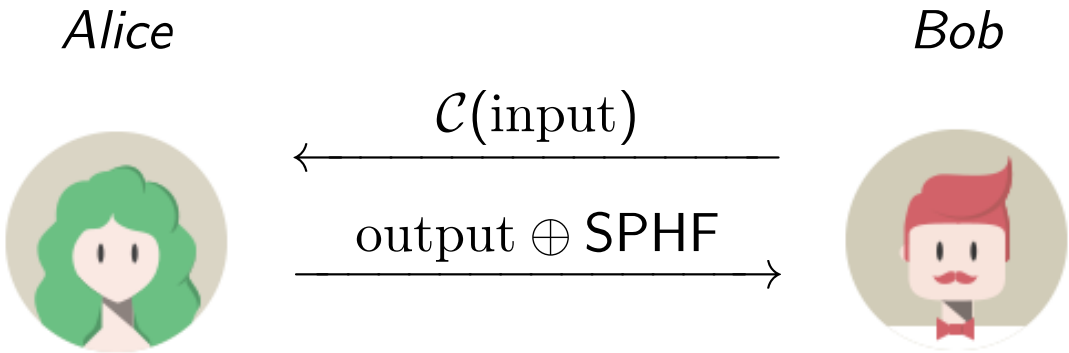
$K = \mathrm{ProjHash}_L(\mathrm{hp}; x, w)$

SPHF

Any encryption of a solution of a NP problem :

- encryption of a password
- encryption of a credential
- solution of an equation
- etc.

Conditional Action



$\rightsquigarrow$ An honest user learns the output.

$\rightsquigarrow$ The server learns nothing.

Oblivious Signature-based envelope [LDB05]

OSBE

Alice the whistle-blower

$\mathcal{C}(\text{cred})$ ←

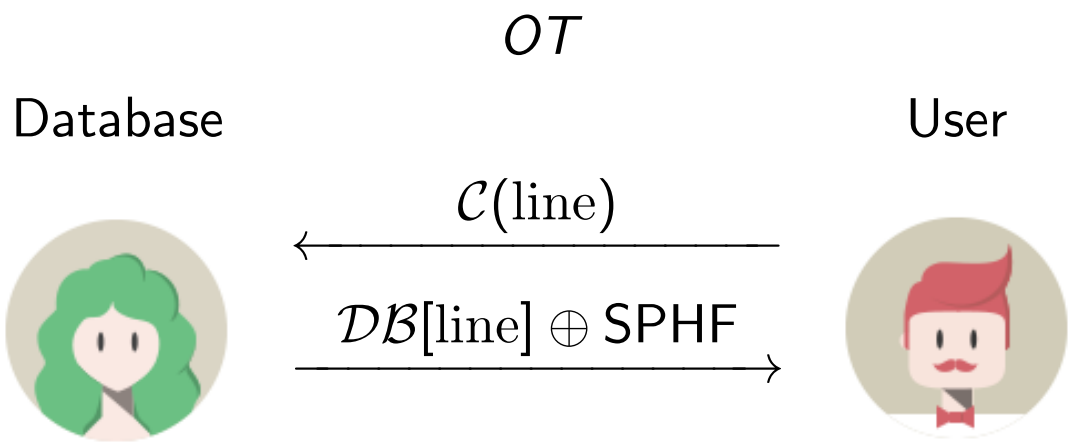
→ $M \oplus \text{SPHF}$

Trustful journalists

⇝ An honest user learns the output iff he possesses the signature.

⇝ The server learns nothing.

Oblivious Transfer [Rab81]



$\rightsquigarrow$ The User learns the value of line but nothing else.

$\rightsquigarrow$ The Database learns nothing.

Authenticated Key Exchange [BM92]

Alice

$\mathsf{SPHF}_A \oplus \mathsf{SPHF}_B$

$\mathcal{C}(\mathrm{pw}_A)$ ←

$\mathcal{C}(\mathrm{pw}_B)$ →

Bob

$\mathsf{SPHF}_B \oplus \mathsf{SPHF}_A$

$\rightsquigarrow$ The Users have the same shared key at the end, if they have the same password

$\rightsquigarrow$ Otherwise they learn nothing

$\rightsquigarrow$ Can be done with other things than password

Authenticated Key Exchange [BM92]

Alice

$\mathrm{SPHF}_A \oplus \mathrm{SPHF}_B$

$\mathcal{C}(\mathrm{pw}_A)$ ←

$\mathcal{C}(\mathrm{pw}_B)$ →

Bob

$\mathrm{SPHF}_B \oplus \mathrm{SPHF}_A$

$\rightsquigarrow$ The Users have the same shared key at the end, if they have the same password

$\rightsquigarrow$ Otherwise they learn nothing

$\rightsquigarrow$ Can be done with other things than password

Thank you

If you are interested in any of these, contact us.
Cryptie: me@cryptie.eu or cryptie@fsfe.org
O.Blazy: olivier.blazy@unilim.fr

Thank you

If you are interested in any of these, contact us.
Cryptie: me@cryptie.eu or cryptie@fsfe.org
O.Blazy: olivier.blazy@unilim.fr
PS: Looking for a PhD student

Sources

Thanks to :

- wordclouds.com for 

(in a home made license, +/- CC-BY...)
- janjf93 for  (in CC0)
- sixsixfive for  (in CC0)
- Phantom Open Emoji maintainers and contributors for  (in CC-BY 3.0)
- the Cnil for  etc. (in CC-BY 3.0)