# Strong Cryptography from Weak Secrets

## Building Efficient PKE and IBE from Distributed Passwords

Xavier Boyen[1] Céline Chevalier[2]

Georg Fuchsbauer[3] David Pointcheval[3]

5 May 2010

[1]Université de Liège, Belgium
[2]Telecom ParisTech, Paris, France
[3]École normale supérieure, Paris, France

# Our Contribution

Abdalla, Boyen, Chevalier, Pointcheval:

Distributed Public-Key Cryptography from Weak Secrets

*PKC 2009*

# Our Contribution

Abdalla, Boyen, Chevalier, Pointcheval:
Distributed Public-Key Cryptography from Weak Secrets
*PKC 2009*

## Extend their results

- DDH $\longrightarrow$ DLIN

  ABCP09 ElGamal encryption
  Ours Linear encryption, identity-based encryption

- Practical simulation-sound NIZKs

  ABCP09 Impractical generic construction or random oracles
  Ours Practical standard-model construction

# Outline

# Outline

# Introduction

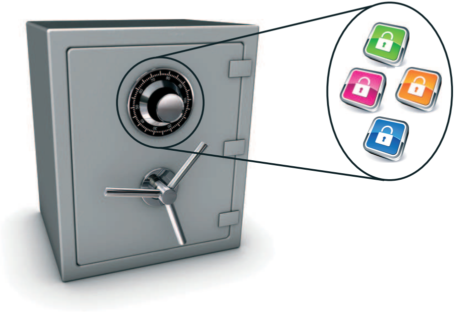
## Goal of distributed cryptography

Base security not on a single person
$\longrightarrow$ Distribute the secret key among several persons

# Introduction

## Goal of distributed cryptography

Base security not on a single person
$\longrightarrow$ Distribute the secret key among several persons
Example: safe with several locks



# Introduction

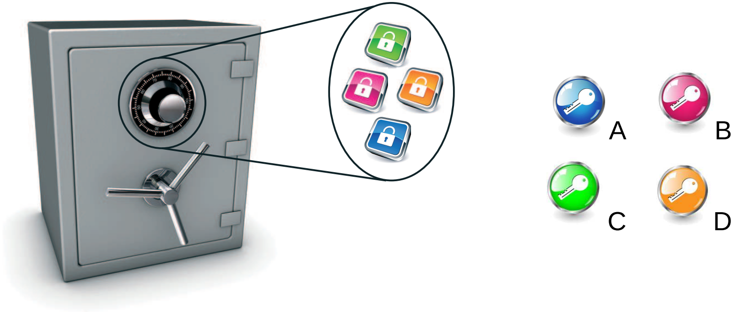
## Goal of distributed cryptography

Base security not on a single person

$\longrightarrow$ Distribute the secret key among several persons

Example: safe with several locks

Every responsable possesses one key



# Introduction

## Goal of distributed cryptography

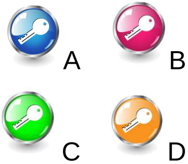
Base security not on a single person

$\longrightarrow$ Distribute the secret key among several persons

Example: safe with several locks

Every responsable possesses one key

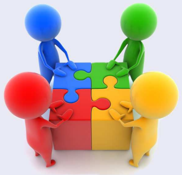
$\longrightarrow$ Presence of *all* responsables necessary



# ElGamal Encryption

## Key distribution

Every player $P_i$ chooses $\mathsf{sk}_i$
(big size and thus high entropy)
$P_i$ publishes $\mathsf{pk}_i = g^{\mathsf{sk}_i}$
Global public key: $\mathsf{pk} = \prod_{i=1}^{n} \mathsf{pk}_i$
Secret key: $\mathsf{sk} = \sum_{i=1}^{n} \mathsf{sk}_i$



# ElGamal Encryption

## Decryption

Every player publishes $\mathrm{pk}_i = g^{\mathrm{sk}_i}$

Global public key: $\mathrm{pk} = \prod_{i=1}^{n} \mathrm{pk}_i$

Secret key: $\mathrm{sk} = \sum_{i=1}^{n} \mathrm{sk}_i$

Parameters: G cyclic, $g$ generator and $h = g^{\mathrm{sk}}$
Cyphertext: $c = E(m;r) = (mh^r, g^r)$

Every player publishes $(g^r)^{\mathrm{sk}_i}$
Multiplying all shares gives $(g^r)^{\mathrm{sk}} = h^r$ thus $mh^r/h^r = m$

# Outline

# Outline

# Introduction

## Disadvantage

Every user must memorize a key of high entropy
$\longrightarrow$ Use passwords

# Introduction

## Disadvantage

Every user must memorize a key of high entropy

$\longrightarrow$ Use passwords

## Passwords in public-key cryptography?

If $\mathsf{pk}_i = g^{\mathsf{pw}_i}$

$\longrightarrow$ Attack by testing every password pw: $g^{\mathsf{pw}} \stackrel{?}{=} \mathsf{pk}_i$

*Offline dictionary attack*

# Introduction

## Disadvantage

Every user must memorize a key of high entropy

$\longrightarrow$ Use passwords

## Passwords in public-key cryptography?

If $\mathsf{pk}_i = g^{\mathsf{pw}_i}$

$\longrightarrow$ Attack by testing every password pw: $g^{\mathsf{pw}} \stackrel{?}{=} \mathsf{pk}_i$

*Offline dictionary attack*

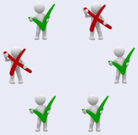
## Best of both worlds

Use *many* passwords to construct distributed key of high entropy

# Distributed Password Public-Key Cryptography

## Model by [ABCP09]

$n$ players $P_1$, ..., $P_n$
One particular player: *group leader*, $P_1$
$n-1$ “mercenaries”, controlled by $P_1$
Every $P_i$ chooses a password $\mathrm{pw}_i$



No assumption of secure channels,
Communication controlled by the adversary
who can *corrupt* players

# Outline

# Universal Composability

## Principle

| Real world | Ideal world |
| --- | --- |

- Protocol

- Ideal Functionality
  - properties of the protocol
  - adversary's goals
  - adversary's means

# Universal Composability

## Principle

| Real world | Ideal world |
| --- | --- |
| • Protocol | • Ideal Functionality<br>  • properties of the protocol<br>  • adversary's goals<br>  • adversary's means |
| • Players | • Virtual players |

# Universal Composability

## Principle

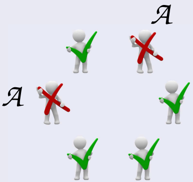
| Real world | Ideal world |
|---|---|
| • Protocol | • Ideal Functionality<br>  ◦ properties of the protocol<br>  ◦ adversary's goals<br>  ◦ adversary's means |
| • Players | • Virtual players |
| • Adversary | • Simulator (to construct) |

Indistinguishability of the two worlds

# Proof principle

## Summary

- There exists an adversary
  - passive or active
  - static or adaptive
  - impersonating players
    with passwords of his choice




- We have to construct a simulator
  plays the role of the virtual players
  that are not corrupted by the adversary
- Simulator does not know passwords chosen by adversary
- The two worlds must be indistinguishable

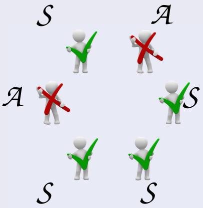
$\longrightarrow$ Need means to *extract* the passwords from the adversary

# Proof principle

## Summary

- There exists an adversary
  - passive or active
  - static or adaptive
  - impersonating players
    with passwords of his choice
- We have to construct a simulator
  plays the role of the virtual players
  that are not corrupted by the adversary
- Simulator does not know passwords chosen by adversary
- The two worlds must be indistinguishable

⟶ Need means to *extract* the passwords from the adversary



# Outline

1. Distributed Cryptography
2. Distributed Password Public-Key Cryptography
   - Introduction
   - Outline of Security Model
   - Construction of Public Key
   - Decryption
3. The Decision-Linear Case

# Ideal Functionality for Public-Key Generation

Parameterized by $\mathtt{PublicKeyGen}$

## Queries allowed to $\mathcal{S}$

- compute $\mathcal{F}$ computes $\mathsf{pk} = \mathtt{PublicKeyGen}(\mathsf{pw}_1, \ldots, \mathsf{pw}_n)$ and sends it to $\mathcal{S}$.
- deliver $\mathcal{F}$ sends pk to player and $\mathcal{S}$

# Instantiation for ElGamal

## Distributed cryptography: public and private key

$n$ players choose $n$ passwords $\mathsf{pw}_i$ $\qquad \mathsf{sk} = \sum_{i=1}^{n} \mathsf{pw}_i \qquad \mathsf{pk} = g^{\mathsf{sk}}$

## Public-key generation

1. first commitment to password (extractable + test)
2. second commitment to password $(g^{\mathsf{pw}_i} h^{r_i}, g^{r_i})$
3. product of commitments: $(g^{\mathsf{sk}} h^{r}, g^{r})$ $\qquad r = \sum r_i$
4. blinding: $(g^{\mathsf{sk}} h^{r}, h) \longrightarrow (g^{\alpha_1 \mathsf{sk}} h^{r\alpha_1}, h^{\alpha_1}) \longrightarrow (g^{\alpha_1\alpha_2 \mathsf{sk}} h^{r\alpha_1\alpha_2}, h^{\alpha_1\alpha_2}) \longrightarrow \cdots \longrightarrow (g^{\alpha \mathsf{sk}} h^{r\alpha}, h^{\alpha})$ $\qquad \alpha = \prod \alpha_i$
5. sending $(h^{\alpha})^{r_i}$: $h^{r\alpha}$ then $g^{\alpha \mathsf{sk}}$
6. unblinding: $g^{\alpha \mathsf{sk}} \longrightarrow g^{\alpha_1 \ldots \alpha_{n-1} \mathsf{sk}} \longrightarrow \cdots \longrightarrow g^{\alpha_1 \mathsf{sk}} \longrightarrow g^{\mathsf{sk}}$

# Outline

# Decryption

## Goal

- One group *leader*
- created public key with help of a group
- wants to decrypt a message (private result)
- secret key is never explicitly computed

Leader wants to compute $c^{\mathrm{sk}}$ from in $:= c$

# Ideal Functionality for Decryption

Parameterized by $\mathtt{PublicKeyVer}$, $\mathtt{SecretKeyGen}$, $\mathtt{PrivateComp}$

## Queries

- Initialization: verify that in and pk are the same for all players
- $\mathtt{PublicKeyVer}(\mathrm{pw}_1, \ldots, \mathrm{pw}_n; \mathrm{pk})$: verification of compatibility of passwords with public key
- compute: $\mathcal{F}$ computes $\mathrm{sk} = \mathtt{SecretKeyGen}(\mathrm{pw}_1, \ldots, \mathrm{pw}_n)$ and $\mathrm{out} = \mathtt{PrivateComp}(\mathrm{sk}, \mathrm{in})$. It informs adversary whether computation succeeded of failed
- leaderDeliver: $\mathcal{F}$ sends out to the *leader* (and the adversary, ie $\mathcal{S}$, if the latter is corrupted)

# Instantiation for ElGamal

## Private decryption of $c$

1. first commitment to passwords (extractable + test)
2. second commitment to passwords $(g^{\mathsf{pw}_i} h^{r_i}, g^{r_i})$
   + commitment $(c^{\mathsf{pw}_i} h^{s_i}, c^{s_i})$
3. blinding/unblinding $\longrightarrow g^{\mathsf{sk}}$ publicly verifiable
4. blinding $\longrightarrow (c^{\alpha \mathsf{sk}} h^{s\alpha}, h^{\alpha})$
5. send $(h^{\alpha})^{s_i} \longrightarrow c^{\alpha \mathsf{sk}}$
6. unblinding: $c^{\alpha \mathsf{sk}} \rightarrow c^{\alpha_1 \ldots \alpha_{n-1} \mathsf{sk}} \rightarrow \cdots \rightarrow c^{\alpha_1 \mathsf{sk}}$ $c^{\mathsf{sk}}$ (private)

# Outline

# Applications

## Identity-Based Encryption (IBE)

- Key generation: system parameters pp
  master secret key sk
- User private key generation (extraction):
  $(\mathrm{pp}, \mathrm{sk}, ID) \longrightarrow d$
- Encryption:
  $(\mathrm{pp}, m, ID) \longrightarrow c$
- Decryption:
  $(\mathrm{pp}, c, d) \longrightarrow m$
- Correctness:
  $\forall\, m, ID$
  $Decrypt\,(\mathrm{pp}, Encrypt\,(\mathrm{pp}, m, ID)\,, Extract\,(\mathrm{pp}, \mathrm{sk}, ID)) = m$

# Applications

## Two IBE schemes

- Password-based Boneh-Franklin IBE [BF01]
  $H(\text{id})$: Hash of the user identity
  compute: $d_{\text{id}} = H(\text{id})^{\text{sk}}$
  $\longrightarrow$ analogous to $c^{\text{sk}}$, similar to ElGamal
- Password-based Boneh-Boyen IBE [BB04]
  compute: $d_{\text{id}} = ({g_0}^{\text{sk}}(g_1^{\text{id}} g_2)^r, g_3^r)$, randomized!
  $\longrightarrow$ new techniques for secret-key functionality with randomness

# Applications

## Two IBE schemes

- Password-based Boneh-Franklin IBE [BF01]
  $H(\mathrm{id})$: Hash of the user identity
  compute: $d_{\mathrm{id}} = H(\mathrm{id})^{\mathrm{sk}}$
  $\longrightarrow$ analogous to $c^{\mathrm{sk}}$, similar to ElGamal
- Password-based Boneh-Boyen IBE [BB04]
  compute: $d_{\mathrm{id}} = ({g_0}^{\mathrm{sk}}(g_1^{\mathrm{id}} g_2)^r, g_3^r)$, randomized!
  $\longrightarrow$ new techniques for secret-key functionality with randomness

Both schemes rely on pairings
$\longrightarrow$ cannot assume DDH

# Changing the Commitments

## Commitment

| El Gamal | | Linear encryption |
|---|---|---|
| $(g^r, g^{\mathrm{pw}}h^r)$ | $\longrightarrow$ | $({g_1}^r, {g_2}^s, g^{\mathrm{pw}}{g_3}^{r+s})$ |

## Improvements

- Efficient zero-knowledge proofs for commitments (Groth-Sahai)
- No need for NIZK proofs for correct blinding and de-blinding
  $h, c^{\mathrm{sk}} \longrightarrow h^\alpha, c^{\alpha\mathrm{sk}}$
  $e(h, c^{\alpha\mathrm{sk}}) = e(h^\alpha, c^{\mathrm{sk}})$

Thank you! ☺