

Strong Cryptography from Weak Secrets

Building Efficient PKE and IBE from Distributed Passwords

Xavier Boyen¹ Céline Chevalier²
Georg Fuchsbauer³ David Pointcheval³

5 May 2010

¹Université de Liège, Belgium

²Telecom ParisTech, Paris, France

³École 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 \rightarrow 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

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

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

Introduction

Goal of distributed cryptography

Base security not on a single person

—→ Distribute the secret key among several persons

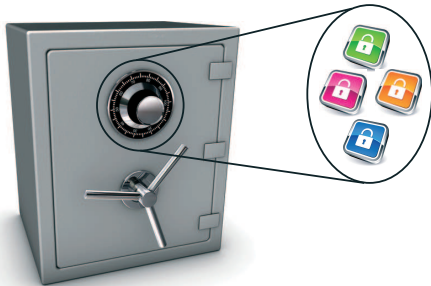
Introduction

Goal of distributed cryptography

Base security not on a single person

→ Distribute the secret key among several persons

Example: safe with several locks



Introduction

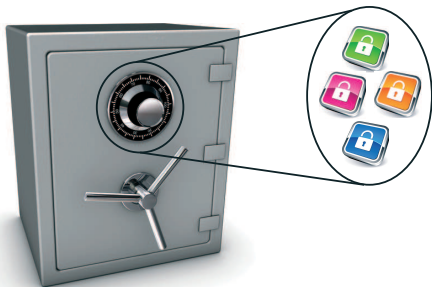
Goal of distributed cryptography

Base security not on a single person

→ Distribute the secret key among several persons

Example: safe with several locks

Every responsible possesses one key



Introduction

Goal of distributed cryptography

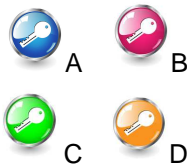
Base security not on a single person

→ Distribute the secret key among several persons

Example: safe with several locks

Every responsible possesses one key

→ Presence of *all* responsables necessary



ElGamal Encryption

Key distribution

Every player P_i chooses sk_i
(big size and thus high entropy)

P_i publishes $pk_i = g^{sk_i}$

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

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



ElGamal Encryption

Decryption

Every player publishes $pk_i = g^{sk_i}$

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

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

Parameters: G cyclic, g generator and $h = g^{sk}$

Cyphertext: $c = E(m; r) = (mh^r, g^r)$

Every player publishes $(g^r)^{sk_i}$

Multiplying all shares gives $(g^r)^{sk} = h^r$ thus $mh^r / h^r = m$

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

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

Introduction

Disadvantage

Every user must memorize a key of high entropy

→ Use passwords

Introduction

Disadvantage

Every user must memorize a key of high entropy

→ Use passwords

Passwords in public-key cryptography?

If $pk_i = g^{pw_i}$

→ Attack by testing every password pw: $g^{pw} \stackrel{?}{=} pk_i$

Offline dictionary attack

Introduction

Disadvantage

Every user must memorize a key of high entropy

→ Use passwords

Passwords in public-key cryptography?

If $pk_i = g^{pw_i}$

→ Attack by testing every password pw: $g^{pw} \stackrel{?}{=} pk_i$

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, \dots, P_n

One particular player: *group leader*, P_1

$n - 1$ “mercenaries”, controlled by P_1

Every P_i chooses a password pw_i

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



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

Universal Composability

Principle

Real world

- Protocol

Ideal world

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

Universal Composability

Principle

Real world

- Protocol
- Players

Ideal world

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

Universal Composability

Principle

Real world

- Protocol
- Players
- Adversary

Ideal world

- Ideal Functionality
 - properties of the protocol
 - adversary's goals
 - adversary's means
- Virtual players
- 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
- Need means to *extract* the passwords from the adversary

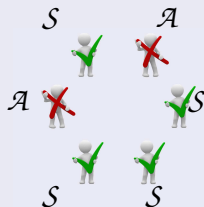


Proof principle

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 `PublicKeyGen`

Queries allowed to \mathcal{S}

- **compute** \mathcal{F} computes $pk = \text{PublicKeyGen}(pw_1, \dots, pw_n)$ 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 pw_i $sk = \sum_{i=1}^n pw_i$ $pk = g^{sk}$

Public-key generation

- 1 first commitment to password (extractable + test)
- 2 second commitment to password ($g^{pw_i} h^{r_i}, g^{r_i}$)
- 3 product of commitments: ($g^{sk} h^r, g^r$) $r = \sum r_i$
- 4 blinding: ($g^{sk} h^r, h$) \rightarrow ($g^{\alpha_1 sk} h^{r\alpha_1}, h^{\alpha_1}$) \rightarrow ($g^{\alpha_1 \alpha_2 sk} h^{r\alpha_1 \alpha_2}, h^{\alpha_1 \alpha_2}$) \rightarrow
 $\dots \rightarrow$ ($g^{\alpha sk} h^{r\alpha}, h^\alpha$) $\alpha = \prod \alpha_i$
- 5 sending (h^α) ^{r_i} : $h^{r\alpha}$ then $g^{\alpha sk}$
- 6 unblinding: $g^{\alpha sk} \rightarrow g^{\alpha_1 \dots \alpha_{n-1} sk} \rightarrow \dots \rightarrow g^{\alpha_1 sk} \rightarrow g^{sk}$

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

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^{sk} from $\text{in} := c$

Ideal Functionality for Decryption

Parameterized by `PublicKeyVer`, `SecretKeyGen`, `PrivateComp`

Queries

- **Initialization**: verify that `in` and `pk` are the same for all players
- `PublicKeyVer(pw1, ..., pwn; pk)`: verification of compatibility of passwords with public key
- **compute**: \mathcal{F} computes `sk = SecretKeyGen(pw1, ..., pwn)` and `out = PrivateComp(sk, in)`. It informs adversary whether computation succeeded or 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 ($g^{pw_i} h^{r_i}, g^{r_i}$)
+ commitment ($c^{pw_i} h^{s_i}, c^{s_i}$)
- 3 blinding/unblinding $\rightarrow g^{sk}$ publicly verifiable
- 4 blinding $\rightarrow (c^{\alpha sk} h^{s\alpha}, h^\alpha)$
- 5 send $(h^\alpha)^{s_i} \rightarrow c^{\alpha sk}$
- 6 unblinding: $c^{\alpha sk} \rightarrow c^{\alpha_1 \dots \alpha_{n-1} sk} \rightarrow \dots \rightarrow c^{\alpha_1 sk}$ c^{sk} (private)

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

Applications

Identity-Based Encryption (IBE)

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

Applications

Two IBE schemes

- Password-based Boneh-Franklin IBE [BF01]

$H(\text{id})$: Hash of the user identity

compute: $d_{\text{id}} = H(\text{id})^{\text{sk}}$

→ analogous to c^{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!

→ new techniques for secret-key functionality with randomness

Applications

Two IBE schemes

- Password-based Boneh-Franklin IBE [BF01]

$H(\text{id})$: Hash of the user identity

compute: $d_{\text{id}} = H(\text{id})^{\text{sk}}$

→ analogous to c^{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!

→ new techniques for secret-key functionality with randomness

Both schemes rely on pairings

→ cannot assume DDH

Changing the Commitments

Commitment

$$\begin{array}{ccc} \text{El Gamal} & \longrightarrow & \text{Linear encryption} \\ (g^r, g^{\text{pw}} h^r) & & (g_1^r, g_2^s, g^{\text{pw}} g_3^{r+s}) \end{array}$$

Improvements

- Efficient zero-knowledge proofs for commitments (Groth-Sahai)
- No need for NIZK proofs for correct blinding and de-blinding

$$\begin{aligned} h, c^{\text{sk}} &\longrightarrow h^\alpha, c^{\alpha\text{sk}} \\ e(h, c^{\alpha\text{sk}}) &= e(h^\alpha, c^{\text{sk}}) \end{aligned}$$

Thank you! 😊