DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

RSA® CONFERENCE 2013
FEBRUARY 25 – MARCH 1
SAN FRANCISCO

Fair Exchange of Short Signatures without Trusted Third Party

Philippe Camacho
University of Chile

Digital Goods Economy

iTunes

Enforcing Secure Transactions through a Trusted Third Party (TTP)

PayPal™

Google™ Checkout

amazon.com®

Problems with TTP

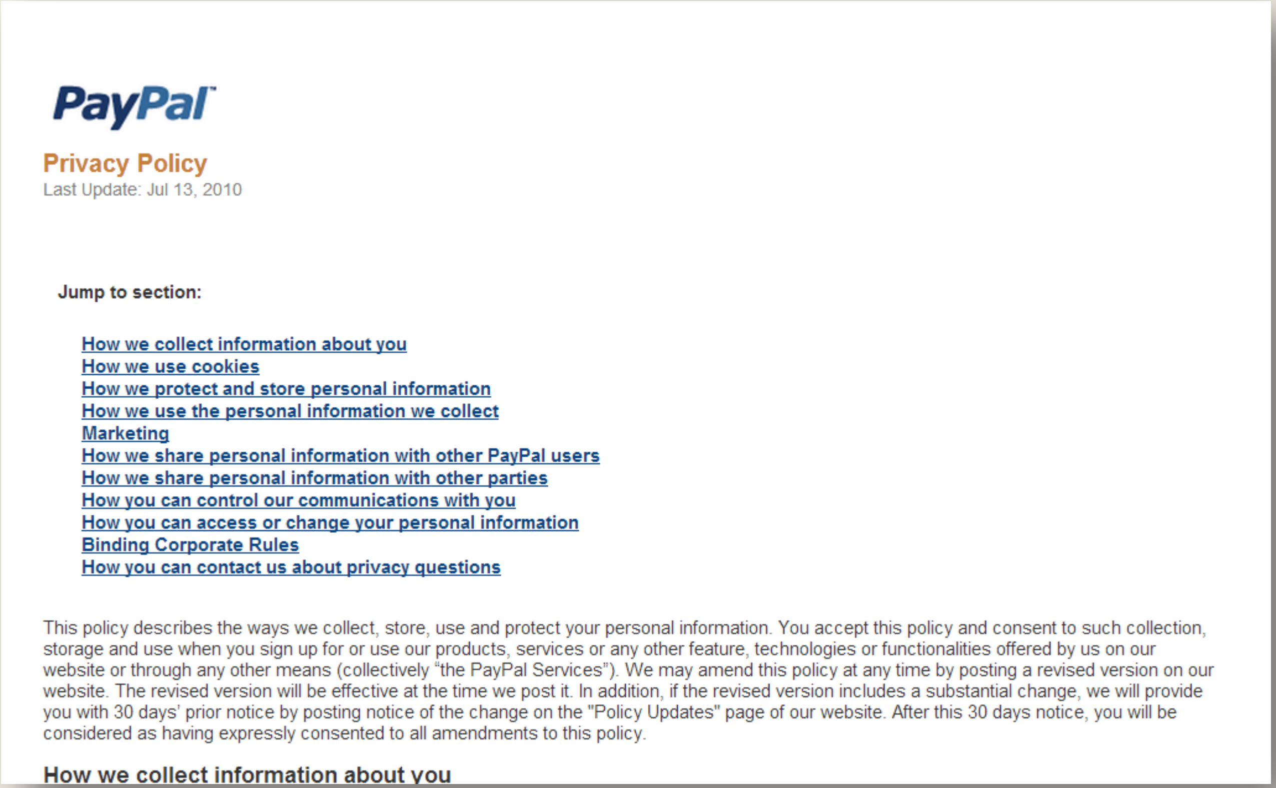
Anonymous Claims To Have Hacked 28,000 PayPal Passwords For Guy Fawkes Day

The Huffington Post | By Cavan Sieczkowski

Posted: 11/05/2012 11:15 am EST Updated: 11/05/2012 1:01 pm EST

Like 829 people like this.

Problems with TTP



PayPal

Privacy Policy

Last Update: Jul 13, 2010

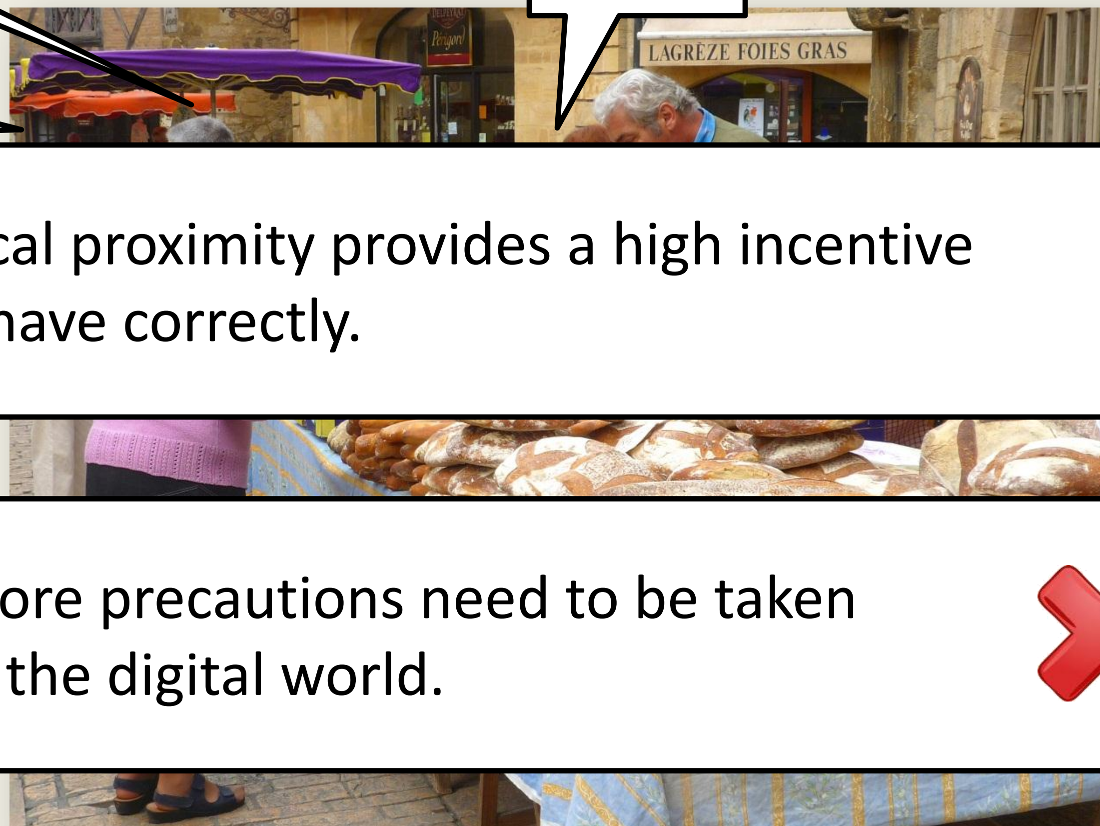
Jump to section:

- How we collect information about you
- How we use cookies
- How we protect and store personal information
- How we use the personal information we collect
- Marketing
- How we share personal information with other PayPal users
- How we share personal information with other parties
- How you can control our communications with you
- How you can access or change your personal information
- Binding Corporate Rules
- How you can contact us about privacy questions

This policy describes the ways we collect, store, use and protect your personal information. You accept this policy and consent to such collection, storage and use when you sign up for or use our products, services or any other feature, technologies or functionalities offered by us on our website or through any other means (collectively "the PayPal Services"). We may amend this policy at any time by posting a revised version on our website. The revised version will be effective at the time we post it. In addition, if the revised version includes a substantial change, we will provide you with 30 days' prior notice by posting notice of the change on the "Policy Updates" page of our website. After this 30 days notice, you will be considered as having expressly consented to all amendments to this policy.

How we collect information about you

Fair Exchange in the Physical World is “easy”



Physical proximity provides a high incentive to behave correctly.

More precautions need to be taken in the digital world.

Modeling Transactions with Digital Signatures

The problem: Who starts first?
Impossibility Result **[Cleve86]**

Software License

Digital Check

Buyer

Seller

Gradual Release of a Secret

Bob’s signature

Alice’s signature

1

Allows to circumvent Cleve’s impossibility result (relaxed security definition).

0

How do I know that the bit I received is not garbage?

Our Construction

- Fair Exchange of Digital Signatures
- Boneh-Boyen [BB04] Short Signatures
- No TTP
- Practical

Contributions

- Formal definition of *Partial Fairness*

- Efficiency

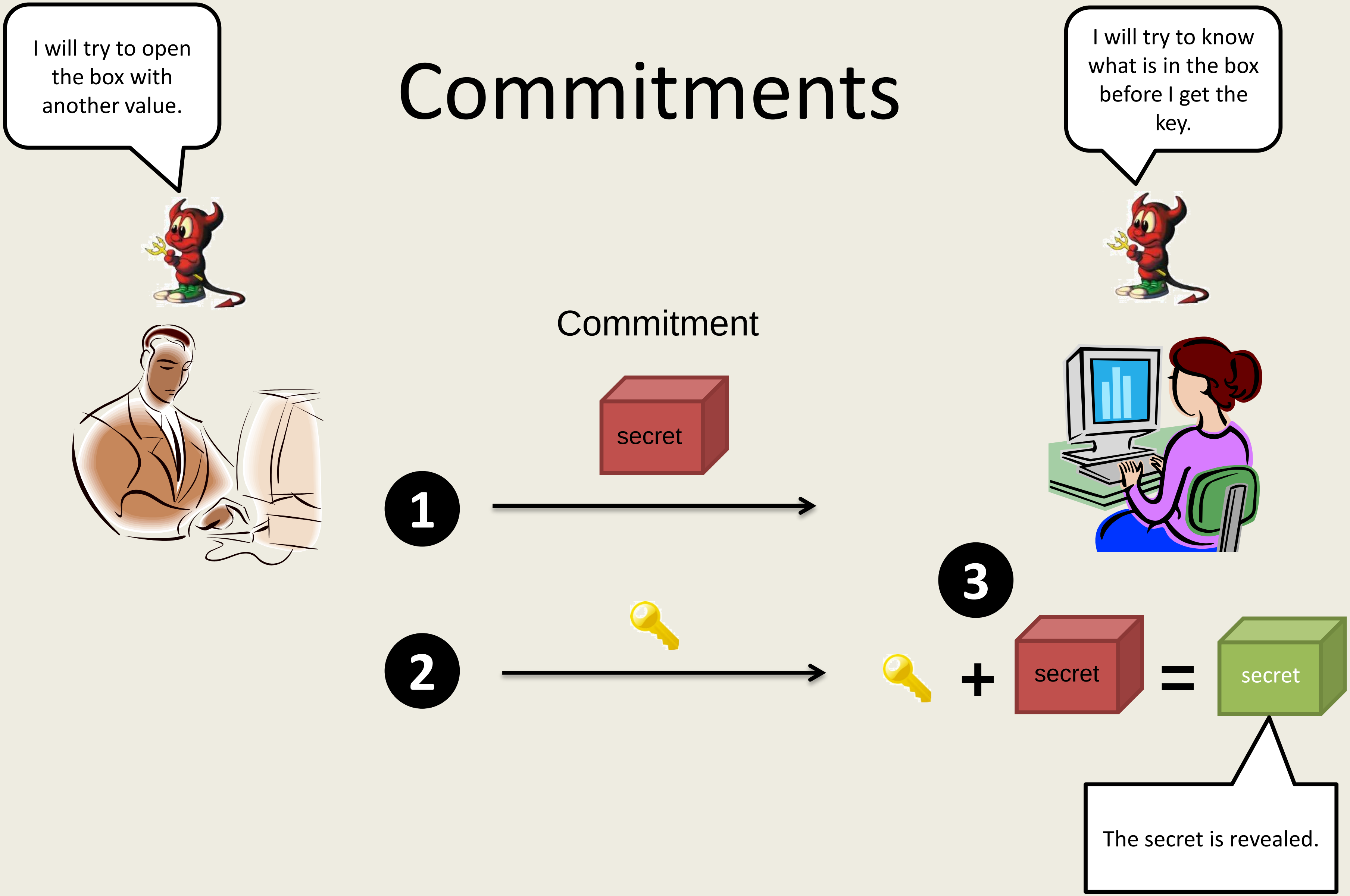
	κ: Security Parameter	$\boldsymbol{\kappa = 160}$
# Rounds	$\kappa + 1$	161
Communication	$16\kappa^2 + 12\kappa$ bits	$\approx$ 52 kB
# Crypto operations per participant	$\approx 30\kappa$	≈ 4800

- First protocol for Boneh-Boyen signatures

Contributions

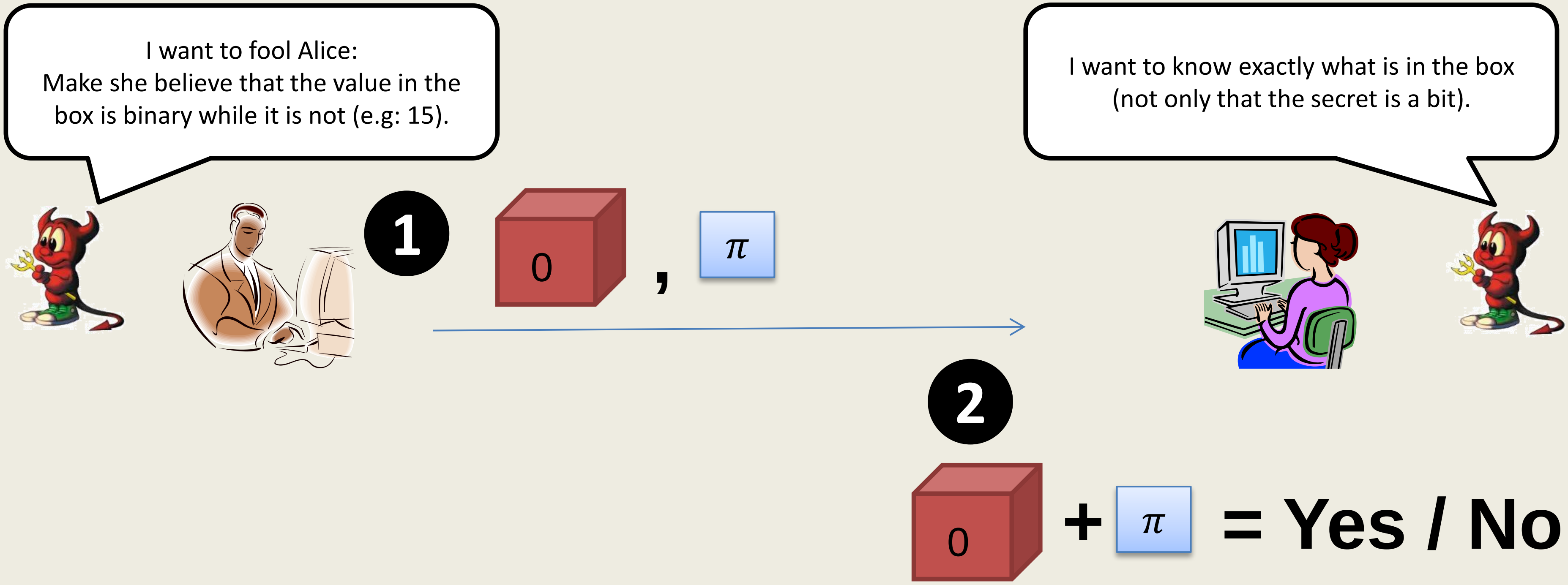
- NIZK argument to prove that a commitment encodes a **bit vector**.
- NIZK argument to prove a commitment to a **bit vector** is the **binary expansion of the discrete logarithm** θ of $D = g^{\theta}$.

Commitments



Non-Interactive
Zero-Knowledge Proofs

Prove something about the secret in the box
without opening the box.



Abstract Protocol

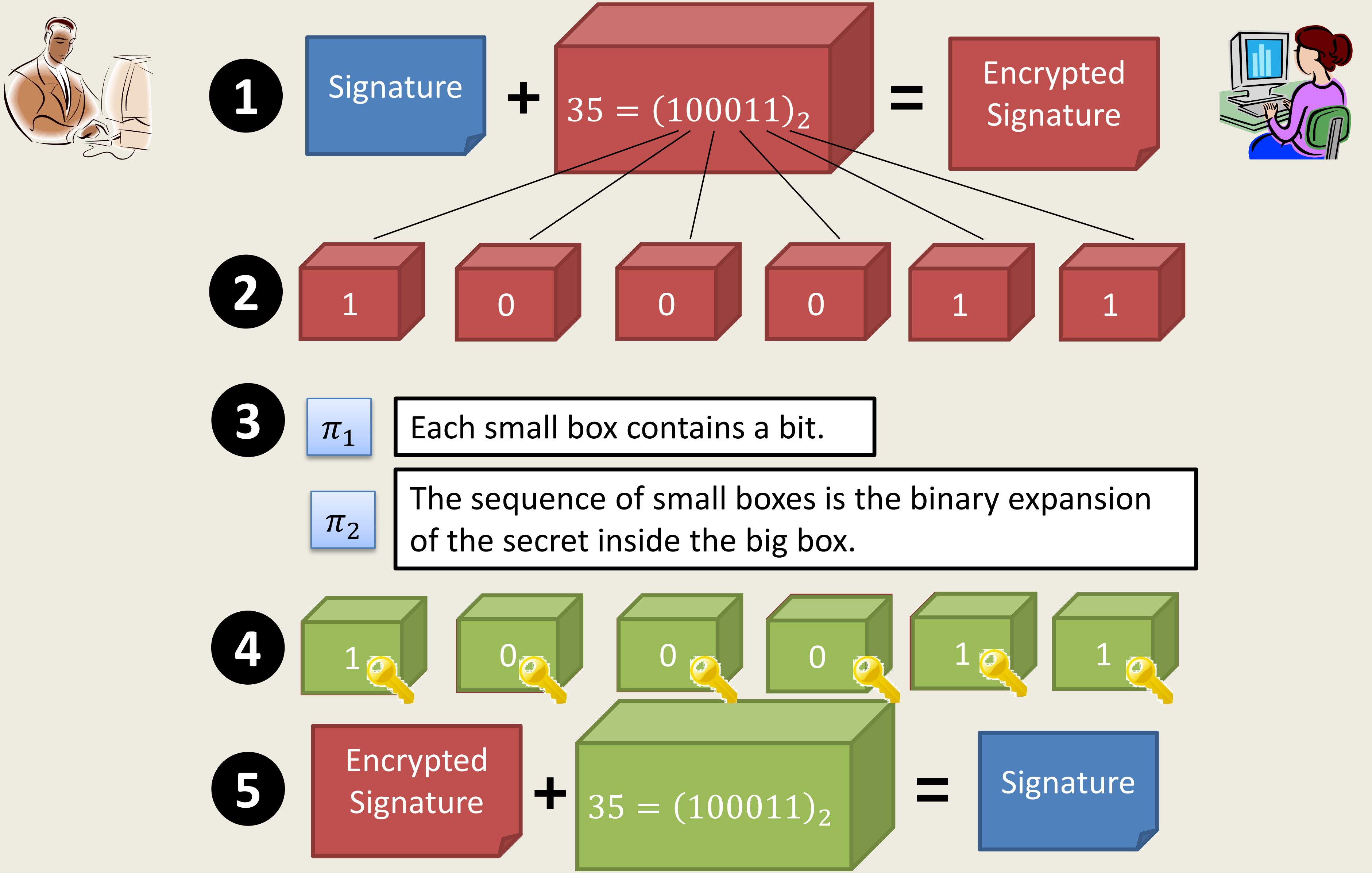
Setup

Step	$\mathcal{P}_A(\mathsf{CRS}, m_A, m_B)$		$\mathcal{P}_B(\mathsf{CRS}, m_A, m_B)$
KeyGen			
1	$(sk_A, pk_A) \leftarrow \mathsf{FEKeyGen}(1^\kappa)$		
2	pk_A	$\longrightarrow$	
3			$(sk_B, pk_B) \leftarrow \mathsf{FEKeyGen}(1^\kappa)$
4		$\longleftarrow$	pk_B
Encrypt Signature			
5	$(\theta_A, \vec{r}_A, \gamma_A) \leftarrow \mathsf{EncSigGen}(\mathsf{CRS}, sk_A, m_A)$		
6	γ_A	$\longrightarrow$	
7			$(\theta_B, \vec{r}_B, \gamma_B) \leftarrow \mathsf{EncSigGen}(\mathsf{CRS}, sk_B, m_B)$
8		$\longleftarrow$	γ_B
Verify Encrypted Signature			
10	$v \leftarrow \mathsf{EncSigCheck}(\mathsf{CRS}, pk_B, m_B, \gamma_B)$		
11	**if** $v = 0$ **then ABORT**		
12			$v \leftarrow \mathsf{EncSigCheck}(\mathsf{CRS}, pk_A, m_A, \gamma_A)$
13			**if** $v = 0$ **then ABORT**
Release Bits			
for $i = 1$ **to** κ:			
14	$\mathsf{open}_{A,i} \leftarrow \mathsf{KeyBitProofGen}(\mathsf{CRS}, \vec{r}_A, \theta_A, i)$		
15	$\mathsf{open}_{A,i}$	$\longrightarrow$	
16			$\mathsf{open}_{B,i} \leftarrow \mathsf{KeyBitProofGen}(\mathsf{CRS}, \vec{r}_B, \theta_B, i)$
17		$\longleftarrow$	$\mathsf{open}_{B,i}$
19	$v_i \leftarrow \mathsf{KeyBitCheck}(\mathsf{CRS}, \mathsf{open}_{B,i}, i)$		
20	**if** $v_i = 0$ **then ABORT**		
21			$v_i \leftarrow \mathsf{KeyBitCheck}(\mathsf{CRS}, \mathsf{open}_{A,i}, i)$
22			**if** $v_i = 0$ **then ABORT**
end for			
Recover Signature			
23	$\sigma_{m_B} \leftarrow \mathsf{EncSigDecrypt}(\gamma_B, \theta_B)$		
24			$\sigma_{m_A} \leftarrow \mathsf{EncSigDecrypt}(\gamma_A, \theta_A)$

Partial Fairness

$$\frac{\Pr[\,\mathsf{SVf}(pk_B, m_B, \sigma_A) = \mathsf{valid}\,]}{\Pr[\,\mathsf{SVf}(pk_A, m_A, \sigma_B) = \mathsf{valid}\,]} \leq Q(\kappa)$$

Protocol


1
Signature
+
$35 = (100011)_2$
=
Encrypted Signature
2
1
0
0
0
1
1
3
π_1
Each small box contains a bit.
π_2
The sequence of small boxes is the binary expansion of the secret inside the big box.
4
1
0
0
0
1
1
5
Encrypted Signature
+
$35 = (100011)_2$
=
Signature

Bilinear maps

- $(p, e, G, G_T, g) \leftarrow BMGen(1^k)$

- $|G| = |G_T| = p$
- $e: G \times G \rightarrow G_T$
- $e(g^a, g^b) = e(g, g)^{ab}$
- $e(g, g)$ generates G_T

Assumptions

- Given $(g, g^s, g^{s^2}, g^{s^3}, \cdots, g^{s^q})$ it's hard to compute
  - $g^{\frac{1}{s}}$ (q- Diffie-Hellman Inversion)
  - $e(g,g,)^{\frac{1}{s}}$ (q-Bilinear Diffie-Hellman Inversion)
  - $(c, g^{\frac{1}{s+c}})$ (q-Strong Diffie-Hellman)
  - $g^{s^{q+i}}$ for $1 \leq i \leq q$ ($q + i$ Diffie-Hellman Exponent)

Assumptions

- **Proposition:** $q - BDHI \Rightarrow\ q + i\ - DHE$

- Our protocol is secure under
  - $q - SDH$
  - $q - BDHI$

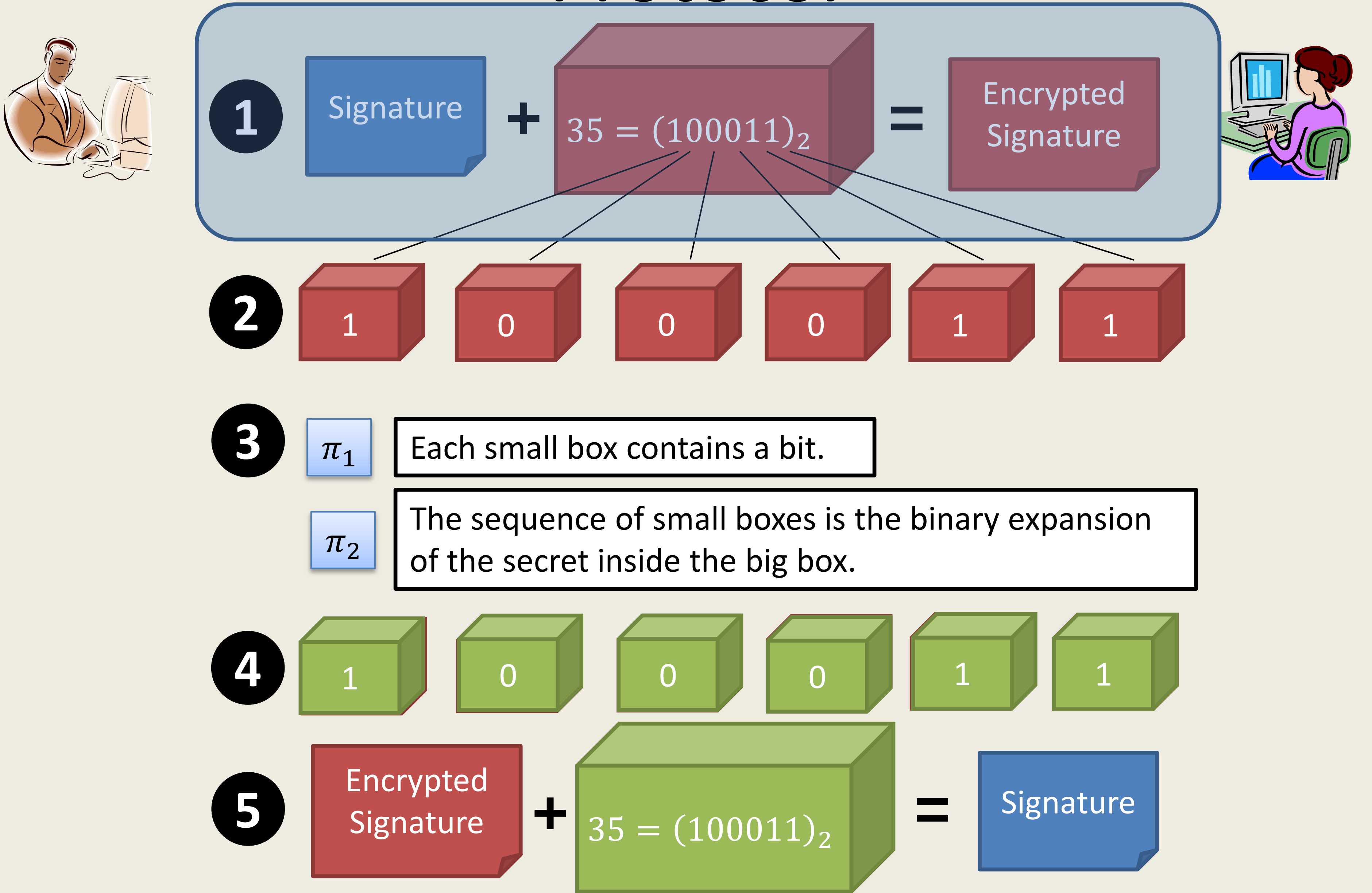
Short Signatures w/o Random Oracle [BB04]

- $\boldsymbol{KeyGen(1^k)}$
  1. $x, y \in Z_p$
  2. $u = g^x, v = g^y$
  3. $pk = (u, v), sk = (x, y)$
  4. return (sk, pk)

- $\boldsymbol{SSign(sk, m)}$
  1. $r \in Z_p$
  2. return $\sigma = (g^{\frac{1}{x+m+yr}}, r) = (\sigma_r, r)$

- $\boldsymbol{SVf(pk, m, \sigma)}$
  1. Check that $\boldsymbol{e(\sigma_r, ug^m v^r)} = e(g^{\frac{1}{x+m+yr}}, g^{x+m+yr}) = \boldsymbol{e(g, g)}$

Protocol


1
Signature
+
35 = (100011)₂
=
Encrypted Signature
2
1
0
0
0
1
1
3
π₁
Each small box contains a bit.
π₂
The sequence of small boxes is the binary expansion of the secret inside the big box.
4
1
0
0
0
1
1
5
Encrypted Signature
+
35 = (100011)₂
=
Signature

The Encrypted Signature

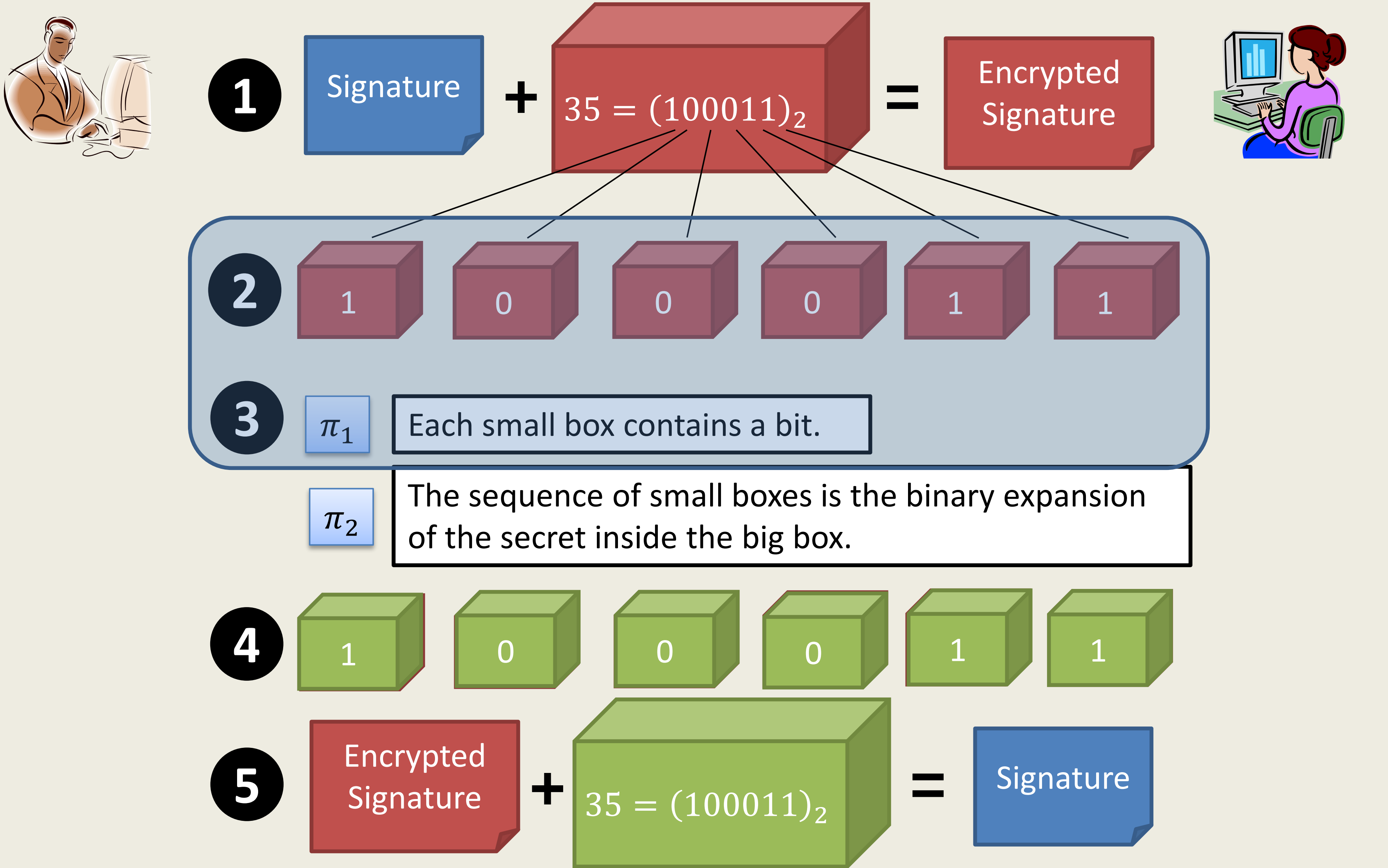
- Computing
  - $\theta \leftarrow Z_p$
  - $D = g^{\theta}$
  - $\boldsymbol{\sigma} = (\boldsymbol{g}^{\frac{\theta}{x+m+yr}}, \boldsymbol{r})$

Secret key / "blinding" factor

Boneh-Boyen signature "blinded" by θ

- Checking
  - Given (D, σ, pk, m) parse σ and pk as
    - $\sigma = (\sigma_\theta, r)$
    - $pk = (g, u = g^x, v = g^y)$
  - $\boldsymbol{e}(\boldsymbol{\sigma_\theta}, \boldsymbol{u}\boldsymbol{g}^{\boldsymbol{m}}\boldsymbol{v}^{\boldsymbol{r}}) = e(g^{\frac{\theta}{\cancel{x+m+yr}}}, g^{\cancel{x+m+yr}}) = \boldsymbol{e}(\boldsymbol{D}, \boldsymbol{g})$

Protocol



1
Signature
+
35 = (100011)₂
=
Encrypted Signature
2
1
0
0
0
1
1
3
π₁
Each small box contains a bit.
π₂
The sequence of small boxes is the binary expansion of the secret inside the big box.
4
1
0
0
0
1
1
5
Encrypted Signature
+
35 = (100011)₂
=
Signature

NIZK argument 1

- $CRS = (g, g^{s}, g^{s^2}, g^{s^3}, \cdots, g^{s^q}) = (g_0, g_1, g_2, g_3, \dots, g_q)$
- **Statement**

  Let $C = (C_1, C_2, \dots, C_q)$

  The prover knows $(r_i, b_i) \in (Z_p \times \{0,1\})$ such that $\boldsymbol{C_i = g^{r_i} g_i^{b_i}}$
- **Argument**
  - $A_i = g_{q-i}^{r_i} g_q^{b_i}$
  - B_i such that $e(A_i, C_i g_i^{-1}) = e(B_i, g)$
  - Return (A_i, B_i) for each $i \in [1..q]$
- **Verification**
  - $e(A_i, g) = e(C_i, g_{q-i})$
  - $e(A_i, C_i g_i^{-1}) = e(B_i, g)$

Shift C_i by $q - i$ positions to the right.

Force the product $\boldsymbol{b_i(b_i - 1)}$ to be computed in the exponent.

NIZK argument 1

- **Theorem:**

  The argument is perfectly complete, computationally sound under the $q + i$ - DHE assumption and perfectly zero-knowledge.

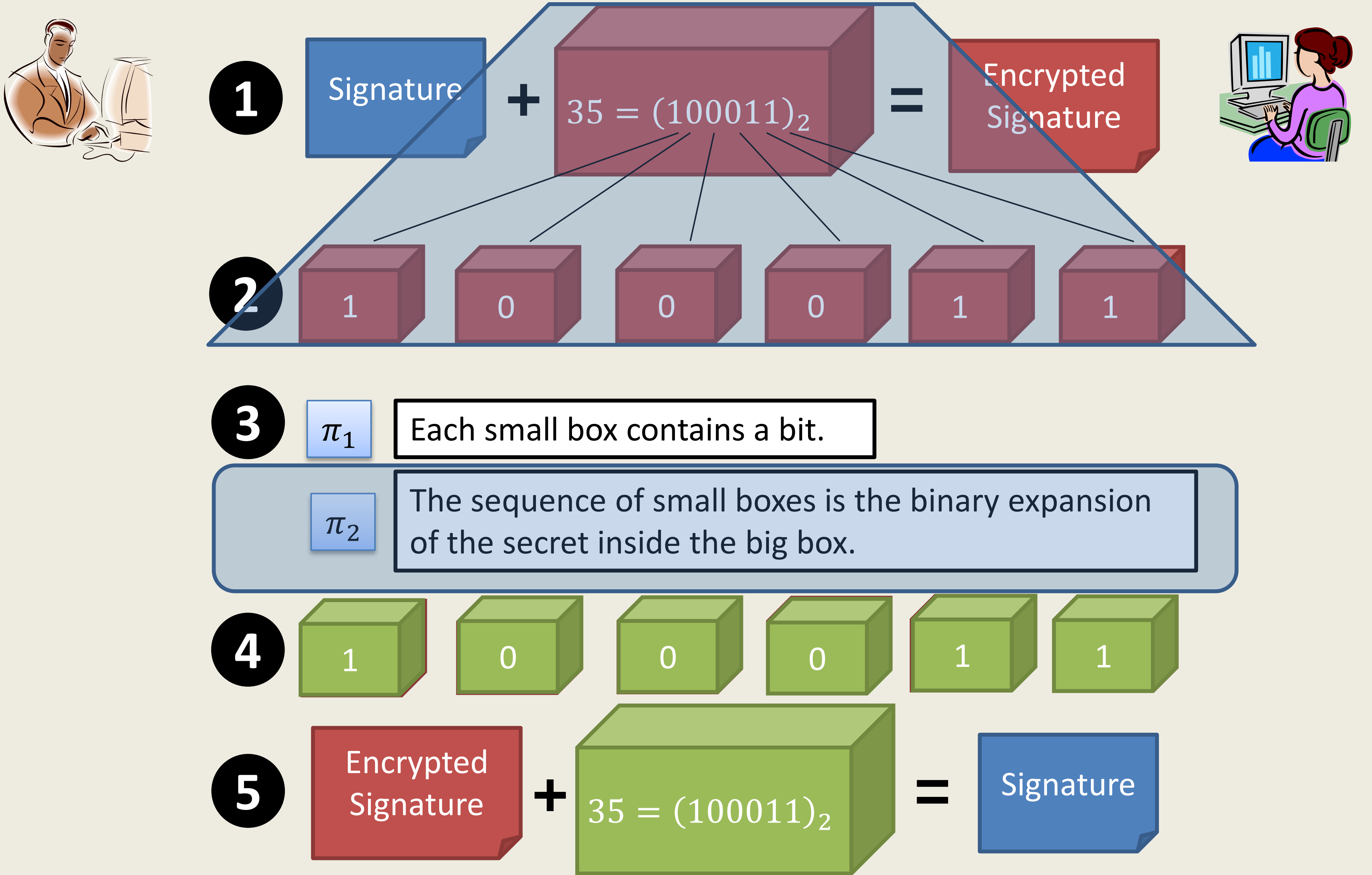
Proof (sketch).

$$e(A_i, C_i g_i^{-1}) = e(g_{q-i}^{r_i} g_q^{b_i}, g^{r_i} g_i^{b_i - 1})$$

$$= e\left(\underbrace{g_{q-i}^{r_i^2} g_q^{r_i(2b_i-1)}}_{B_i} \boldsymbol{g_{q+i}^{b_i(b_i-1)}}, g\right) = e(B_i, g)$$

If $b_i \notin \{0,1\}$, the adversary breaks the $q + i -$ DHE assumption.

Protocol


1
Signature
+
35 = (100011)₂
=
Encrypted Signature
2
1
0
0
0
1
1
3
π1
Each small box contains a bit.
π2
The sequence of small boxes is the binary expansion of the secret inside the big box.
4
1
0
0
0
1
1
5
Encrypted Signature
+
35 = (100011)₂
=
Signature

NIZK argument 2

- $CRS = \left(g, g^{s}, g^{s^2}, g^{s^3}, \cdots, g^{s^q}\right) = \left(g_0, g_1, g_2, g_3, \ldots, g_q\right)$

- We set $q = \kappa$ (security parameter)

- **Statement**
  - The prover knows $(r_i, b_i) \in \ (Z_p \times \ \{0,1\})$ and θ such that $C_i = g^{r_i} g_i^{b_i}$, $D = g^{\theta}$ and

$$\theta = \sum_{i=1}^{\kappa} b_i 2^{i-1}$$

NIZK argument 2

- **Verification:** Input (C

  - Parse $\pi = (r', U, V)$

  - Check that $e(\frac{\prod_{i=1}^{k} C_i}{g^{r'}}, g) = e(U, g_1)$

  - Check that $e(\frac{U}{D}, g) = e(V, g_1 g^{-2})$

$$\prod_{i=1}^{k} C_i = \prod_{i=1}^{k} g^{r_i} g_i^{b_i} \Leftrightarrow [r', b_1, b_2, \ldots, b_\kappa]$$

$$U = \left(\prod_{i=1}^{k} g_i^{b_i}\right)^{1/s} = \prod_{i=1}^{k} g_{i-1}^{b_i} \Leftrightarrow [b_1, b_2, \ldots, b_\kappa]$$

$$r' = \sum_i r_i$$

θ

$U \Leftrightarrow P(s)$ (i.e. $U = g^{P(s)}$)

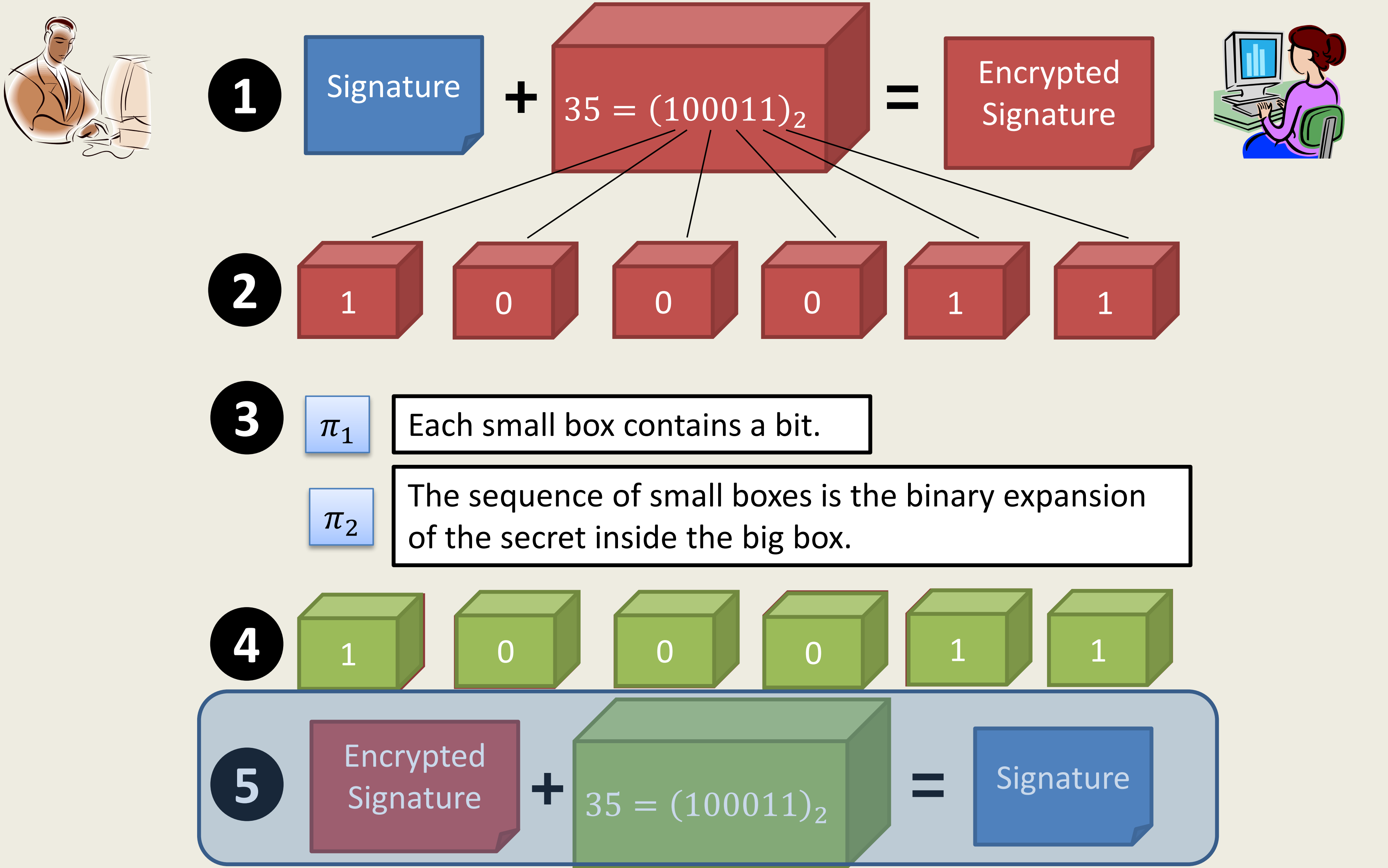
$V \Leftrightarrow W(s)$ s.t. $P(s) - P(2) = W(s)(s-2)$

NIZK argument 2

- **Theorem:**

  The argument is perfectly complete, computationally sound under the $q-SDH$ assumption and perfectly zero-knowledge.

Protocol


1
Signature
+
$35 = (100011)_2$
=
Encrypted Signature
2
1
0
0
0
1
1
3
π_1
Each small box contains a bit.
π_2
The sequence of small boxes is the binary expansion of the secret inside the big box.
4
1
0
0
0
1
1
5
Encrypted Signature
+
$35 = (100011)_2$
=
Signature

Recovering the Signature

- All the bits b_i are revealed
- Compute $\theta = \sum_{i=1}^{\kappa} b_i \, 2^{i-1}$
- We have $\sigma = \left(g^{\frac{\cancel{\theta}}{x+m+yr}}, r\right) = (\sigma_\theta, r)$
- Compute $\boldsymbol{\sigma} = (\sigma_\theta{}^{\mathbf{1/\theta}}, \boldsymbol{r})$

Proofs of Knowledge

- Discrete logarithm θ of
  - $D = g^{\theta}$
- r_i, b_i such that
  - $C_i = g^{r_i} g_i^{b_i}$

Needed in order to simulate the adversary despite it aborts early.

Simultaneous Hardness of Bits for Discrete Logarithm

Holds in the generic group model **[Schnorr98]**

An adversary cannot distinguish between a **random sequence** of $\boldsymbol{\kappa - l}$ bits and the **first** $\boldsymbol{\kappa - l}$ bits **of** $\boldsymbol{\theta}$ given $\boldsymbol{g^{\theta}}$.

$$Adv^{SHDL}(\mathcal{A}, \kappa) = \left|\Pr\left[\begin{array}{c}\theta \stackrel{R}{\leftarrow} \mathbb{Z}_p : \\ 1 \leftarrow \mathcal{A}(g^{\theta}, \theta[1..\kappa - l])\end{array}\right] - \Pr\left[\begin{array}{c}\theta, \alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p : \\ 1 \leftarrow \mathcal{A}(g^{\theta}, \alpha[1..\kappa - l])\end{array}\right]\right|$$

$l = \omega(\log \kappa)$

Conclusion

- Fair exchange protocol for short signatures [BB04] without TTP
- Practical
- Two new NIZK arguments

Thank you!

Partial Fairness

Only contract signing

- A randomized protocol for signing contracts [EGL85]
- Gradual release of a secret [BCDB87]
- Practically and Provably secure release of a secret and exchange of signatures [Damgard95]
- Resource Fairness and Composability of Cryptographic protocols [GMPY06]

RSA, Rabin, ElGamal signatures

"Time-line" assumptions, Generic construction

- **Theorem:**

  The protocol is partially fair under the $\kappa - SDH$ and the $\kappa - BDHI$ assumption.

Proof (Sketch)

- Type I
  - Does not forge values but aborts «early»
  - => He has to break the signature scheme

  - **Careful:**
    What happens if A detects he is simulated?
    - The simulator will try to break the SHDL assumption
    - If few bits remain, it does not win, everything is OK!

Proof (Sketch)

- Type II
  - Forge values
  - The simulator can extract all values computed by adversary and break the soundness of the NIZK arguments or binding property of commitment scheme.

Fully Secure Attribute-Based Systems with Short Ciphertexts/Signatures and Threshold Access Structures

Cheng Chen[1] Jie Chen[2] Hoonwei Lim[2] Zhenfeng Zhang[1]
Dengguo Feng[1] San Ling[2] Huaxiong Wang[2]

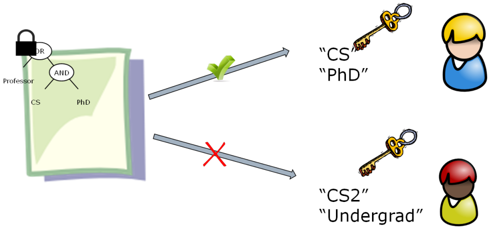
[1]Institute of Software, Chinese Academy of Sciences, China

[2]Nanyang Technological University, Singapore

CT-RSA 2013

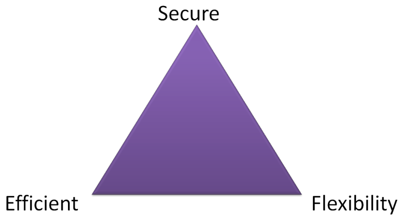
Attribute-based systems [SW05,GPSW06,MPR11]

- Policies and credentials are labeled with attributes
- Highly expressive, fine grained access policy
- Non-interactive role based access control



Performance tradeoff

- Efficiency: communication, computation costs
- Security: adaptive vs selective, CPA vs CCA
- Flexibility: expressiveness



Current status

- Most existing ABE and ABS schemes have linear-size ciphertexts and signatures.
- Some recent proposals focused on reducing the overhead, but achieved better efficiency at the expense of weaker security.
- None work achieve both adaptive security and constant-size ciphertexts and signatures for a relatively expressive access policy.

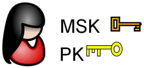
The motive of this work: full security and constant-size overhead

Offer solutions that achieve both full security and constant-size ABE ciphertexts or ABS signatures:

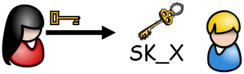
- Give formal definitions and security models for predicate encryption (PE) and predicate signatures (PS).
- Propose a generic construction of attribute-based systems supporting threshold access policies from inner-product systems.
- The resulting attribute-based constructions preserve the properties from underlying inner-product schemes.
- Present concrete constructions of fully secure ABE/ABS with constant-size ciphertexts/signatures from the IPE/IPS schemes tailored to our needs.

Background: predicate encryption (PE)

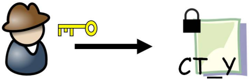
- Setup$(1^\kappa) \rightarrow (PP, Msk)$



- KeyGen$(PP, Msk, X) \rightarrow sk_X$



- Enc$(PP, Y, Msg) \rightarrow CT_Y$



- Dec$(PP, sk_X, CT) \rightarrow Msg'$

$$Dec(PP, sk_X, Enc(PP, Y, Msg)) = Msg \quad \Longleftrightarrow \quad R(X, Y) = 1$$

Security: ciphertext indistinguishability

Experiment $Exp^{ind}_{\mathcal{PE}}(\kappa)$:

$$\boxed{Y \longleftarrow \mathcal{A}}$$

$$b \xleftarrow{R} \{0,1\}$$

$$\mathsf{PP}, \mathsf{MSK} \xleftarrow{R} \mathbf{Setup}$$

$$(Msg_0, Msg_1, Y) \xleftarrow{R} \mathcal{A}^{\mathbf{KeyGen}(\cdot)}(\mathsf{PP})$$

$$\mathsf{CT} \xleftarrow{R} \mathbf{Enc}(\mathsf{PP}, Y, Msg_b)$$

$$b' \longleftarrow \mathcal{A}^{\mathbf{KeyGen}(\cdot)}(\mathsf{PP}, \mathsf{CT})$$

If $b = b'$ and $R(X, Y) \neq 1$ return 1 else return 0

Variants of PE

There exist many public key primitives that can be viewed as special cases of PE:

- ABE: ciphertext-policy (CP) & key-policy (KP)

$$X :\longrightarrow S \subseteq \{att_1, \ldots, att_n\}, \quad Y :\longrightarrow \phi,\ \phi \textit{ is an access structure}$$

$$R(X,Y) = \begin{cases} 1 & \textit{if} \quad S \in \phi \\ 0 & \textit{if} \quad S \notin \phi \end{cases}$$

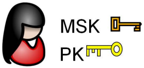
- Inner-product encryption (IPE):

$$X :\longrightarrow \vec{v} \in \mathbb{Z}_p^n, \quad Y :\longrightarrow \vec{x} \in \mathbb{Z}_p^n$$

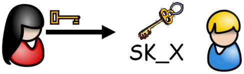
$$R(X,Y) = \begin{cases} 1 & \textit{if} \quad \langle \vec{v}, \vec{x} \rangle = 0 \\ 0 & \textit{if} \quad \langle \vec{v}, \vec{x} \rangle \neq 0 \end{cases}$$

Predicate signature (PS)

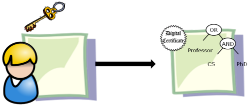
- Setup$(1^\kappa) \to (PP, Msk)$



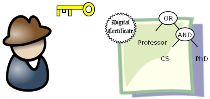
- KeyGen$(PP, Msk, X) \to sk_X$



- Sign$(PP, Y, sk_X, Msg) \to \sigma$



- Verify$(PP, \sigma, Y) \to \{0, 1\}$



$$Verify(PP, Sign(PP, KeyGen(PP, Msk, X), Msg), Y) = 1 \iff R(X, Y) = 1$$

Security: unforgeability

Experiment $Exp^{unf}_{\mathcal{PS}}(\kappa)$:

$Y \longleftarrow \mathcal{A}$

$\mathsf{PP}, \mathsf{MSK} \xleftarrow{R} \textbf{Setup}$

$(Msg, Y, \sigma) \xleftarrow{R} \mathcal{A}^{\textbf{KeyGen}(\cdot),\textbf{Sign}(\cdot)}(\mathsf{PP})$

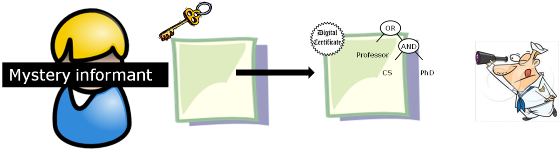
If $\textbf{Verify}(PP, \sigma, Y) = 1, R(X, Y) \neq 1$

and (Msg, Y) has not been made as

signature queries return 1 else return 0

Security: perfect privacy

A predicate signature ensures the verifier only knows that the signer's role can satisfy the specified signing policy.



For any Msg, X_1, X_2 and Y such that $R(X_1, Y) = R(X_2, Y) = 1$, we have

$$\textbf{Sign}(\mathsf{PP}, \textbf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, X_1), Y, Msg) \equiv \textbf{Sign}(\mathsf{PP}, \textbf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, X_2), Y, Msg)$$

Variants of PS

There exist many signature primitives that can be viewed as special cases of PS:

- ABS:

$$X :\longrightarrow S \subseteq \{att_1, \ldots, att_n\}, \quad Y :\longrightarrow \phi,\ \phi \textit{ is an access structure}$$

$$R(X,Y) = \begin{cases} 1 & \textit{if} \quad S \in \phi \\ 0 & \textit{if} \quad S \notin \phi \end{cases}$$

- Inner-product signature (IPS):

$$X :\longrightarrow \vec{v} \in \mathbb{Z}_p^n, \quad Y :\longrightarrow \vec{x} \in \mathbb{Z}_p^n$$

$$R(X,Y) = \begin{cases} 1 & \textit{if} \quad \langle \vec{v}, \vec{x} \rangle = 0 \\ 0 & \textit{if} \quad \langle \vec{v}, \vec{x} \rangle \neq 0 \end{cases}$$

Intuitions of generic constructions: exact threshold policy [KSW08]

Express an attribute subset S as a vector $\vec{x}_S$:

$$\vec{x}_S := (\overbrace{b_1}^{att_1}, \ldots, \overbrace{b_i}^{att_i}, \ldots), \quad for \quad i = 1, 2, \ldots \quad b_i = \begin{cases} 1 & if \quad att_i \in S \\ 0 & if \quad att_i \notin S \end{cases}$$

If S_1 and S_2 have t attributes overlap, we have

$$\langle \vec{x}_{S_1}, \vec{x}_{S_2} \rangle = t$$

Exact threshold policy from inner-product policy

- **Setup**(κ, U): $\mathsf{IPE.Setup}(\kappa, n+1) \rightarrow (\mathsf{PP}, \mathsf{MSK})$;
- **Enc**$(\mathsf{PP}, \Gamma := (\Omega, t), Msg)$: $\mathsf{IPE.Enc}(\mathsf{PP}, (t, \vec{x}_\Omega), M) \rightarrow \mathsf{CT}_\Gamma$;
- **KeyGen**$(\mathsf{PP}, \mathsf{MSK}, S)$: $\mathsf{IPE.KeyGen}(\mathsf{PP}, \mathsf{MSK}, (-1, \vec{x}_S)) \rightarrow \mathsf{SK}_S$;
- **Dec**$(\mathsf{PP}, \mathsf{CT}_\Gamma, \mathsf{SK}_S)$: $\mathsf{IPE.Dec}(\mathsf{PP}, \mathsf{CT}_\Gamma, \mathsf{SK}_S) \rightarrow Msg$.

Correctness. $\langle(-1, \vec{x}_S), (t, \vec{x}_\Omega)\rangle = 0$ if $|\Omega \cap S| = t$.

Exact threshold to threshold: IPE to tKP-ABE

Introduce multiple IPE secret keys to achieve flexibility:

$$\mathsf{tKP.KeyGen}(\mathsf{PP}, \Gamma := (\Omega, t), \mathsf{MSK}):$$

$$\mathsf{IPE.KeyGen}(\mathsf{PP}, (t, \vec{x}_\Omega), \mathsf{MSK}) \to \mathsf{IPE.SK}_1$$

$$\mathsf{IPE.KeyGen}(\mathsf{PP}, (t+1, \vec{x}_\Omega), \mathsf{MSK}) \to \mathsf{IPE.SK}_2$$

$$\mathsf{IPE.KeyGen}(\mathsf{PP}, (t+2, \vec{x}_\Omega), \mathsf{MSK}) \to \mathsf{IPE.SK}_3$$

$$\vdots$$

$$\mathsf{KP.SK}_{(\Omega,t)} := \{\mathsf{IPE.SK}_j\}_{1 \le j \le m-t+1}$$

$$\mathsf{tKP.Enc}(\mathsf{PP}, S, Msg):$$

$$\mathsf{IPE.Enc}(\mathsf{PP}, (-1, \vec{x}_S), Msg) \to \mathsf{CT}$$

Exact threshold to threshold: IPE to tCP-ABE

$\text{tCP.KeyGen}(\text{PP}, S, \text{MSK}):$

$\text{IPE.KeyGen}(\text{PP}, (1, \vec{x}_S, 0), \text{MSK}) \rightarrow \text{IPE.SK}_1$

$\text{IPE.KeyGen}(\text{PP}, (1, \vec{x}_S, -1), \text{MSK}) \rightarrow \text{IPE.SK}_2$

$\text{IPE.KeyGen}(\text{PP}, (1, \vec{x}_S, -2), \text{MSK}) \rightarrow \text{IPE.SK}_3$

$\vdots$

$\text{CP.SK}_S := \{\text{IPE.SK}_i\}_{1 \leq i \leq |S|-1}$

$\text{tCP.Enc}(\text{PP}, \Gamma := (\Omega, t), Msg):$

$\text{IPE.Enc}(\text{PP}, (-t, \vec{x}_\Omega, 1), Msg) \rightarrow \text{CT}$

Exact threshold to threshold: IPS to tABS

$$\mathsf{tABS.KeyGen}(\mathsf{PP}, S, \mathsf{MSK}):$$

$$\mathsf{IPS.KeyGen}(\mathsf{PP}, (1, \vec{x}_S, 0), \mathsf{MSK}) \to \mathsf{IPS.SK}_1$$

$$\mathsf{IPS.KeyGen}(\mathsf{PP}, (1, \vec{x}_S, -1), \mathsf{MSK}) \to \mathsf{IPS.SK}_2$$

$$\mathsf{IPS.KeyGen}(\mathsf{PP}, (1, \vec{x}_S, -2), \mathsf{MSK}) \to \mathsf{IPS.SK}_3$$

$$\vdots$$

$$\mathsf{ABS.SK}_S := \{\mathsf{IPS.SK}_i\}_{1 \le i \le |S|-1}$$

$$\mathsf{tABS.Sign}(\mathsf{PP}, \mathsf{ABS.SK}_S, \Gamma := (\Omega, t), Msg):$$

$$\mathsf{IPS.Sign}(\mathsf{PP}, \mathsf{IPS.SK}_{k-t+1}, (-t, \vec{x}_\Omega, 1), Msg) \to \sigma$$

where $\mathsf{IPS.SK}_{k-t+1} \leftarrow \mathsf{IPS.KeyGen}(\mathsf{PP}, (-t, \vec{x}_S, t-k), \mathsf{MSK})$

$k := |S \cap \Omega| \ge t$

Concrete constructions of tABE and tABS

Basing the transformation from inner-product systems to attribute-based systems supporting threshold access structures:

- Properties-preserving:
  - full security/selective security
  - constant-size ciphertext/signature
  - perfect privacy
- Building blocks of IPE/IPS schemes tailored to our needs:
  - IPE: [AL10], but too complicated.
  - IPS: non-existent.

The properties of underlying IPE & IPS

scheme	group order	based on	size of CT or signature
[AL10]	prime	none	constant
Our IPE	composite	[AL10]	constant
Our IPS1	composite	our IPE	constant
Our IPS2	prime	our IPE & DPVS	constant

Our IPE: fully secure IPE with constant-size ciphertexts in composite order group

- $\mathsf{IPE.Setup}(\lambda, n) \to (\mathsf{PP}, \mathsf{MSK})$
  $\mathsf{PP} := \left(\mathcal{I} := (N = p_1 p_2 p_3, G, G_T, e), g, \vec{h} := (h_0, \ldots, h_n), e(g,g)^\alpha\right)$
  $\mathsf{MSK} := (\alpha, \boxed{X_3})$.
- $\mathsf{IPE.KeyGen}(\mathsf{PP}, \mathsf{MSK}, \vec{v}) \to \mathsf{IPE.SK}_{\vec{v}} := (K_0, K_1, \ldots, K_n)$

$$K_0 := g^r \cdot \boxed{R_0}, \quad K_1 := g^\alpha h_0^r \cdot \boxed{R_1}, \quad \left\{ K_i := \left(h_1^{-\frac{v_i}{v_1}} h_i \right)^r \cdot \boxed{R_i} \right\}_{i=2,\ldots,n}.$$

- $\mathsf{IPE.Enc}(\mathsf{PP}, \vec{x}, Msg) \to \mathsf{CT} := (C, C_0, C_1)$

$$C := Msg \cdot e(g,g)^{\alpha s}, \quad C_0 := g^s, \quad C_1 := \left(h_0 \prod_{j=1}^{n} h_j^{x_j} \right)^s.$$

- $\mathsf{IPE.Dec}(\mathsf{PP}, \vec{x}, \mathsf{IPE.SK}_{\vec{v}}, \mathsf{CT})$: The algorithm computes

$$Msg' = C \cdot \frac{e(C_1, K_0)}{e(C_0, K_1 \prod_{j=2}^{n} K_j^{x_j})}$$

The security of our IPE & IPS

- Dual system proof [Wat09] is applied to obtain full security.
- Some composite order complexity assumptions are introduced.
- Our IPS scheme is prefectly private because the distribution of the signature is the same.

Comparisons

	scheme	security	size of SK	size of CT or Sig	expressiveness	Pai
CP-ABE	[EM+09]	selective	$\mathcal{O}(n)$	$\mathcal{O}(1)$	(n,n)-threshold	2
	[CZF11]	selective	$\mathcal{O}(n)$	$\mathcal{O}(1)$	and-gate	2
	[HLR10]	selective	$\mathcal{O}(n)$	$\mathcal{O}(1)$	threshold	3
	[GZC11]	selective	$\mathcal{O}(n)^2$	$\mathcal{O}(1)$	threshold	3
	[OT10]	full	$\mathcal{O}(n)$	$\mathcal{O}(n)$	general	$\mathcal{O}(n)$
	Our CP-ABE	full	$\mathcal{O}(n)^2$	$\mathcal{O}(1)$	threshold	2
KP-ABE	[ABP11]	selective	$\mathcal{O}(n)^2$	$\mathcal{O}(1)$	general	3
	[OT10]	full	$\mathcal{O}(n)$	$\mathcal{O}(n)$	general	$\mathcal{O}(n)$
	Our KP-ABE	full	$\mathcal{O}(n)^2$	$\mathcal{O}(1)$	threshold	2
ABS	[HLLR12a]	selective	$\mathcal{O}(n)$	$\mathcal{O}(1)$	threshold	12
	[HLLR12b]	selective	$\mathcal{O}(n)^2$	$\mathcal{O}(1)$	threshold	3
	[OT11]	full	$\mathcal{O}(n)$	$\mathcal{O}(n)$	general	$\mathcal{O}(n)$
	Our ABS	full	$\mathcal{O}(n)^2$	$\mathcal{O}(1)$	threshold	3

Conclusion

- We define the syntax and security notions of PE/PS.
- We bridge a connection between inner-product systems and attribute-based systems.
- Our tABE/tABS schemes achieve both full security and short ciphertexts/signatures.