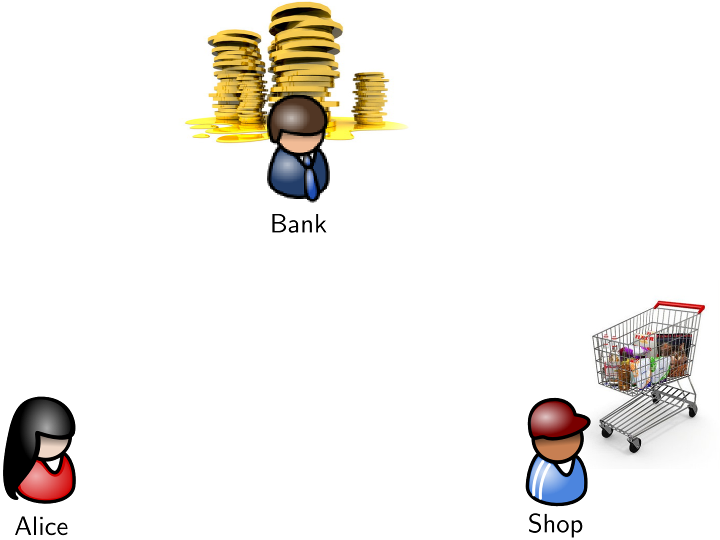
Primitives et constructions cryptographiques pour la confiance numrique

Damien Vergnaud

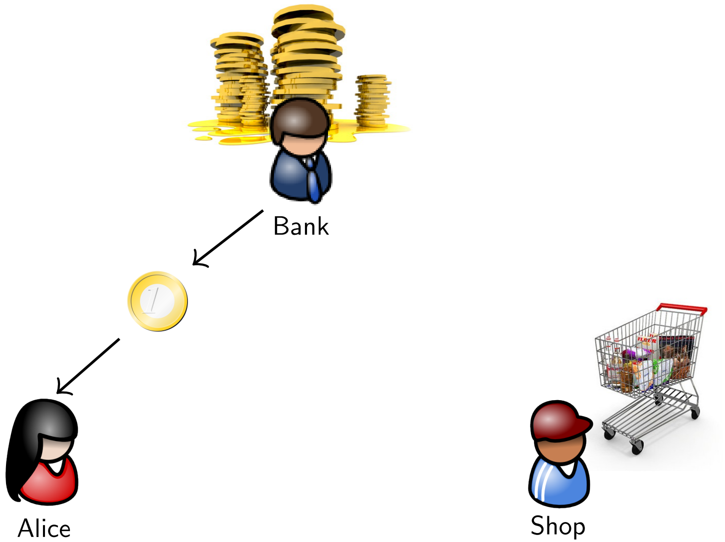
École normale supérieure – C.N.R.S. – I.N.R.I.A.

3 avril 2014

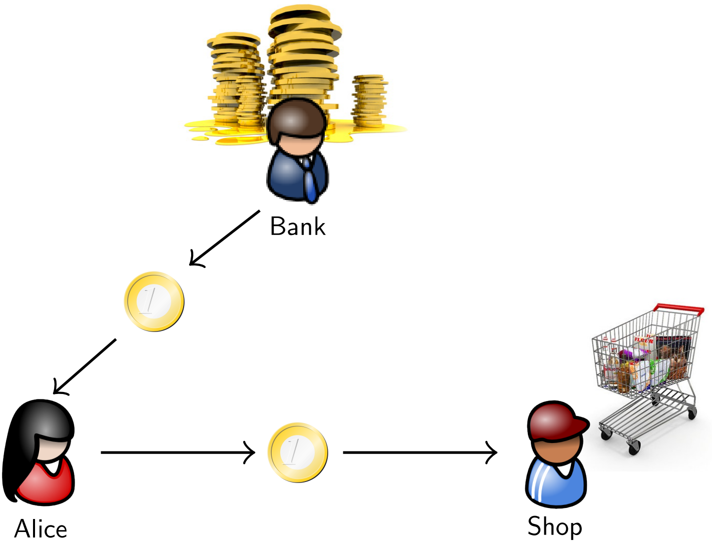
Motivation: The Concept of E-cash



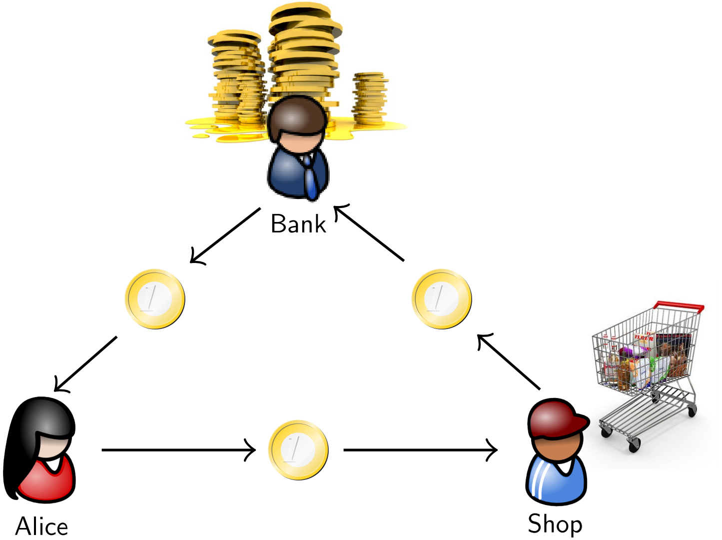
Motivation: The Concept of E-cash



Motivation: The Concept of E-cash



Motivation: The Concept of E-cash



Desirable Properties of E-cash

- **Off-line:** bank **not** present at the time of payment
- **Traceability of double spenders:**
  each time a user spends a coin more than once he will be detected
- **Anonymity:** if a user does not spend a coin twice, she remains anonymous
- **Fairness:** perfect anonymity enables perfect crimes
  $\rightsquigarrow$ an authority can trace coins that were acquired illegally.
- **Transferability:** received e-cash can be spend without involving the bank
  - fundamental property of regular cash
  - Chaum and Pederson (1992) $\rightsquigarrow$ impossible without increasing the coin size

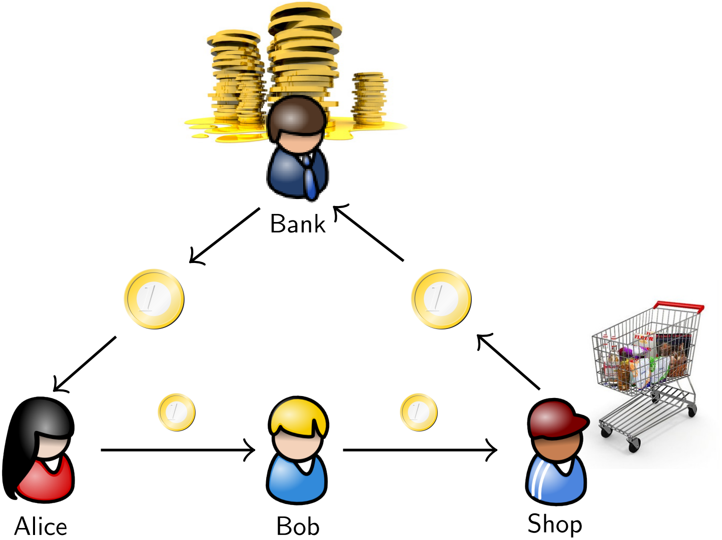
Desirable Properties of E-cash

- **Off-line:** bank **not** present at the time of payment
- **Traceability of double spenders:**
  each time a user spends a coin more than once he will be detected
- **Anonymity:** if a user does not spend a coin twice, she remains anonymous
- **Fairness:** perfect anonymity enables perfect crimes
  $\rightsquigarrow$ an authority can trace coins that were acquired illegally.
- **Transferability:** received e-cash can be spend without involving the bank
  - fundamental property of regular cash
  - Chaum and Pederson (1992) $\rightsquigarrow$ impossible without increasing the coin size

Desirable Properties of E-cash

- **Off-line:** bank **not** present at the time of payment
- **Traceability of double spenders:**
  each time a user spends a coin more than once he will be detected
- **Anonymity:** if a user does not spend a coin twice, she remains anonymous
- **Fairness:** perfect anonymity enables perfect crimes
  $\rightsquigarrow$ an authority can trace coins that were acquired illegally.
- **Transferability:** received e-cash can be spend without involving the bank
  - fundamental property of regular cash
  - Chaum and Pederson (1992) $\rightsquigarrow$ impossible without increasing the coin size

The Concept of Transferable E-cash



Contents

Zero-Knowledge Proof Systems

- Goldwasser, Micali and Rackoff introduced interactive **zero-knowledge proofs** in 1985
  - the paper was rejected a couple of times
  - ...then they won the Gödel award for it

$\rightsquigarrow$ proofs that reveal nothing other than the validity of assertion being proven

- Central tool in study of cryptographic protocols
  - **Anonymous credentials**
  - **Online voting**
  - ...

Zero-Knowledge Proof Systems

- Goldwasser, Micali and Rackoff introduced interactive **zero-knowledge proofs** in 1985
  - the paper was rejected a couple of times
  - ...then they won the Gödel award for it

$\rightsquigarrow$ proofs that reveal nothing other than the validity of assertion being proven

- Central tool in study of cryptographic protocols
  - **Anonymous credentials**
  - **Online voting**
  - ...

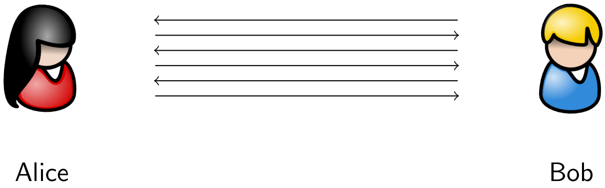
Zero-Knowledge Proof Systems

- Goldwasser, Micali and Rackoff introduced interactive **zero-knowledge proofs** in 1985
  - the paper was rejected a couple of times
  - . . . then they won the Gödel award for it

$\rightsquigarrow$ proofs that reveal nothing other than the validity of assertion being proven

- Central tool in study of cryptographic protocols
  - **Anonymous credentials**
  - **Online voting**
  - . . .

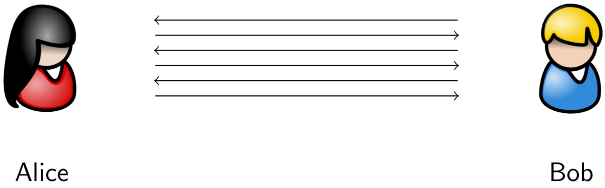
Zero-knowledge Interactive Proof



- interactive method for one party to prove to another that a statement $\mathcal{S}$ is true, without revealing anything other than the veracity of $\mathcal{S}$.

1. **Completeness:** $\mathcal{S}$ is true $\rightsquigarrow$ verifier will be convinced of this fact
2. **Soundness:** $\mathcal{S}$ is false $\rightsquigarrow$ no cheating prover can convince the verifier that $\mathcal{S}$ is true
3. **Zero-knowledge:** $\mathcal{S}$ is true $\rightsquigarrow$ no cheating verifier learns anything other than this fact. (weaker version: **Witness indistinguishability**)

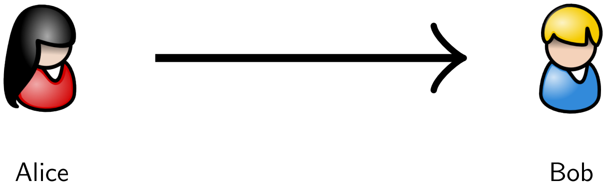
Zero-knowledge Interactive Proof



- interactive method for one party to prove to another that a statement $\mathcal{S}$ is true, without revealing anything other than the veracity of $\mathcal{S}$.

1. **Completeness:** $\mathcal{S}$ is true $\rightsquigarrow$ verifier will be convinced of this fact
2. **Soundness:** $\mathcal{S}$ is false $\rightsquigarrow$ no cheating prover can convince the verifier that $\mathcal{S}$ is true
3. **Zero-knowledge:** $\mathcal{S}$ is true $\rightsquigarrow$ no cheating verifier learns anything other than this fact. (weaker version: **Witness indistinguishability**)

Non-interactive Zero-knowledge Proof



- non-interactive method for one party to prove to another that a statement $\mathcal{S}$ is true, without revealing anything other than the veracity of $\mathcal{S}$.

1. **Completeness:** $\mathcal{S}$ is true $\rightsquigarrow$ verifier will be convinced of this fact
2. **Soundness:** $\mathcal{S}$ is false $\rightsquigarrow$ no cheating prover can convince the verifier that $\mathcal{S}$ is true
3. **Zero-knowledge:** $\mathcal{S}$ is true $\rightsquigarrow$ no cheating verifier learns anything other than this fact. (weaker version: **Witness indistinguishability**)

History of NIZK Proofs

Inefficient NIZK

- Blum-Feldman-Micali, 1988.
- Damgard, 1992.
- Killian-Petrank, 1998.
- Feige-Lapidot-Shamir, 1999.
- De Santis-Di Crescenzo-Persiano, 2002.

Alternative: Fiat-Shamir heuristic transforms interactive ZK proof into NIZK
But there are examples of insecure Fiat-Shamir transformation

- Groth-Ostrovsky-Sahai, 2006.
- Groth-Sahai, 2008.

History of NIZK Proofs

Inefficient NIZK

- Blum-Feldman-Micali, 1988.
- Damgard, 1992.
- Killian-Petrank, 1998.
- Feige-Lapidot-Shamir, 1999.
- De Santis-Di Crescenzo-Persiano, 2002.

Alternative: Fiat-Shamir heuristic transforms interactive ZK proof into NIZK
But there are examples of insecure Fiat-Shamir transformation

- Groth-Ostrovsky-Sahai, 2006.
- Groth-Sahai, 2008.

History of NIZK Proofs

Inefficient NIZK

- Blum-Feldman-Micali, 1988.
- Damgard, 1992.
- Killian-Petrank, 1998.
- Feige-Lapidot-Shamir, 1999.
- De Santis-Di Crescenzo-Persiano, 2002.

Alternative: Fiat-Shamir heuristic transforms interactive ZK proof into NIZK
But there are examples of insecure Fiat-Shamir transformation

- Groth-Ostrovsky-Sahai, 2006.
- Groth-Sahai, 2008.

History of NIZK Proofs

Inefficient NIZK

- Blum-Feldman-Micali, 1988.
- Damgard, 1992.
- Killian-Petrank, 1998.
- Feige-Lapidot-Shamir, 1999.
- De Santis-Di Crescenzo-Persiano, 2002.

Alternative: Fiat-Shamir heuristic transforms interactive ZK proof into NIZK
But there are examples of insecure Fiat-Shamir transformation

- **Groth-Ostrovsky-Sahai, 2006.**
- Groth-Sahai, 2008.

Applications of NIZK Proofs

- Fancy signature schemes
  - group signatures
  - ring signatures
  - ...
- Efficient non-interactive proof of correctness of shuffle
- Non-interactive anonymous credentials
- CCA-2-secure encryption schemes
- Identification
- E-cash
- ...

Composite order bilinear structure: What ?

$(e, \mathbb{G}, \mathbb{G}_T, g, n)$ bilinear structure:

- $\mathbb{G}$, $\mathbb{G}_T$ multiplicative groups of order $n = pq$
  - $n =$ **RSA integer**
- $\langle g \rangle = \mathbb{G}$
- $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$
  - $\langle e(g, g) \rangle = \mathbb{G}_{\mathbb{T}}$
  - $e(g^a, g^b) = e(g, g)^{ab}$, $a, b \in \mathbb{Z}$
- $\left.\begin{array}{l}\text{deciding group membership,}\\ \text{group operations,}\\ \text{bilinear map}\end{array}\right\}$ efficiently computable.

Composite order bilinear structure: How ?

- Groups are instantiated using supersingular elliptic curves E over finite fields $\mathbb{F}_\ell$, ℓ mod $-1(modn)$ prime.

- Groups are very large: $N \geq 2^{2048}$ to prevent factoring attack.

- Pairings are **slow**:

usual pairing-based crypto (prime-order curve)	$\mathbb{G} \subset E(\mathbb{F}_\ell) \simeq$ **256 bits** $\mathbb{G}_T \subset \mathbb{F}^*_{\ell^6} \simeq$ **2048 bits** **3 ms** pairing
composite-order groups (supersingular curve)	$\mathbb{G} \subset E(\mathbb{F}_\ell) \simeq$ **2048 bits** $\mathbb{G}_T \subset \mathbb{F}^*_{\ell^2} \simeq$ **4096 bits** **150 ms** pairing

Conclusion: composite-order elliptic curves negates many advantages of ECC

Composite order bilinear structure: Why ?

1. **Deciding Diffie-Hellman tuples:** given $(g, g^a, g^b, g^c) \in \mathbb{G}^4$

$$c = ab \iff e(g^a, g^b) = e(g, g^c)$$

2. **If $\mathbf{h^q = 1}$:** for all $v \in \mathbb{G}$

$$e(h, v)^q = 1$$

$$e(g^a h^b, g)^q = e(g, g)^a$$

Applications: "Somewhat homomorphic" encryption, Traitor tracing, Ring and group signatures, Attribute-based encryption, Fully secure HIBE, ...

Composite order bilinear structure: Why ?

1. **Deciding Diffie-Hellman tuples:** given $(g, g^a, g^b, g^c) \in \mathbb{G}^4$

$$c = ab \iff e(g^a, g^b) = e(g, g^c)$$

2. **If $\mathbf{h^q = 1}$:** for all $v \in \mathbb{G}$

$$e(h, v)^q = 1$$

$$e(g^a h^b, g)^q = e(g, g)^a$$

Applications: "Somewhat homomorphic" encryption, Traitor tracing, Ring and group signatures, Attribute-based encryption, Fully secure HIBE, ...

Composite order bilinear structure: Why ?

1. **Deciding Diffie-Hellman tuples:** given $(g, g^a, g^b, g^c) \in \mathbb{G}^4$

$$c = ab \iff e(g^a, g^b) = e(g, g^c)$$

2. **If $\mathbf{h^q = 1}$:** for all $v \in \mathbb{G}$

$$e(h, v)^q = 1$$

$$e(g^a h^b, g)^q = e(g, g)^a$$

Applications: "Somewhat homomorphic" encryption, Traitor tracing, Ring and group signatures, Attribute-based encryption, Fully secure HIBE, ...

Boneh-Goh-Nissim Encryption Scheme

Public key: $(e, \mathbb{G}, \mathbb{G}_T, n)$ bilinear structure with $n = pq$
$g, h \in \mathbb{G}$ with $\mathrm{ord}(h) = q$.

Secret key: p, q

Encryption: $c = g^m h^r \ (r \xleftarrow{R} \mathbb{Z}_n)$

Decryption: $c^q = (g^m h^r)^q = g^{mq} h^{qr} = (g^q)^m$ (+ discrete log)

IND-CPA-secure under the:

Subgroup Membership Assumption

Hard to distinguish $h \in \mathbb{G}$ of order q from random h of order n

Boneh-Goh-Nissim Commitment Scheme

Public key: $(e, \mathbb{G}, \mathbb{G}_T, n)$ bilinear structure with $n = pq$
$g, h \in \mathbb{G}$ with $\mathrm{ord}(h) = q$.

Commitment: $c = g^m h^r \ (r \xleftarrow{R} \mathbb{Z}_n)$

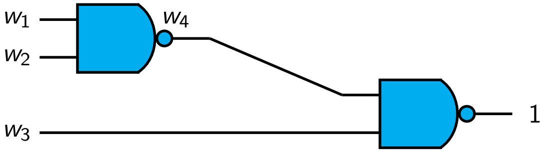
- **Perfectly binding:** unique m mod p
- **Computationally hiding:** indistinguishable from h of order n
- **Addition:** $(g^a h^r) \cdot (g^b h^s) = g^{a+b} h^{r+s}$
- **Multiplication:**

$$\begin{aligned} e(g^a h^r, g^b h^s) &= e(g^a, g^b) e(h^r, g^b) e(g^a, h^s) e(h^r, h^s) \\ &= e(g, g)^{ab} e(h, g^{as+rb} h^{rs}) \end{aligned}$$

Groth-Ostrovsky-Sahai: NIZK Proof for Circuit SAT

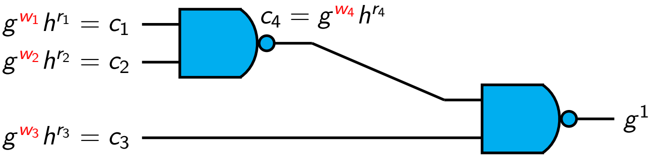
- Groth, Ostrovsky and Sahai (2006)
  - Perfect completeness, perfect soundness, computational zero-knowledge for NP
  - Common reference string: $O(k)$ bits
  - Proof: $O(|C|k)$ bits

- Circuit-SAT is **NP-complete**



- **Idea:**
  - Commit w_i using BGN encryption
  - Prove the validity using homomorphic properties

NIZK Proof for Circuit SAT



- Prove $w_i \in \{0,1\}$ for $i \in \{1,2,3,4\}$
- Prove $w_4 = \neg(w_1 \wedge w_2)$
- Prove $1 = \neg(w_3 \wedge w_4)$

Proof for c Containing 0 or 1

- $w \bmod p \in \{0,1\} \iff w(w-1) = 0 \bmod p$
- For $c = g^{w} h^{r}$ we have

$$\begin{aligned} e(c, cg^{-1}) &= e(g^{w} h^{r}, g^{w-1} h^{r}) \\ &= e(g^{w}, g^{w-1}) e(h^{r}, g^{w-1}) e(g^{w}, h^{r}) e(h^{r}, h^{r}) \\ &= e(g,g)^{w(w-1)} e(h, (\underbrace{g^{2w-1} h^{r}}_{\pi})^{r}) \end{aligned}$$

- $\pi = g^{2w-1} h^{r}$ = proof that c contains 0 or 1 $\bmod p$.
(c detemines w uniquely $\bmod p$ since $\mathrm{ord}(h) = q$)
- Randomizable proof !

Proof for c Containing 0 or 1

- $w \bmod p \in \{0,1\} \iff w(w-1) = 0 \bmod p$
- For $c = g^w h^r$ we have

$$\begin{aligned} e(c, cg^{-1}) &= e(g^w h^r, g^{w-1} h^r) \\ &= e(g^w, g^{w-1}) e(h^r, g^{w-1}) e(g^w, h^r) e(h^r, h^r) \\ &= e(g,g)^{w(w-1)} e(h, (\underbrace{g^{2w-1} h^r}_{\pi})^r) \end{aligned}$$

- $\pi = g^{2w-1} h^r$ = proof that c contains 0 or 1 $\bmod p$.
(c detemines w uniquely $\bmod p$ since $\mathrm{ord}(h) = q$)
- **Randomizable proof !**

A Simple Observation

b_0	b_1	b_2	$b_0 + b_1 + 2b_2 - 2$
0	0	0	-2
0	0	1	0
0	1	0	-1
0	1	1	1
1	0	0	-1
1	0	0	-1
1	0	1	1
1	1	0	0
1	1	1	2

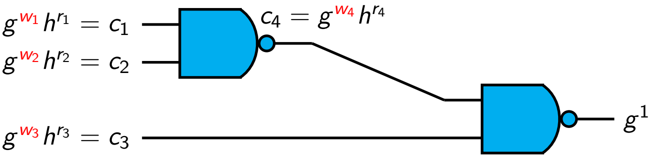
$$b_2 = \neg(b_0 \wedge b_1) \iff b_0 + b_1 + 2b_2 - 2 \in \{0, 1\}$$

A Simple Observation

b_0	b_1	b_2	$b_0 + b_1 + 2b_2 - 2$
0	0	0	-2
0	0	1	0
0	1	0	-1
0	1	1	1
1	0	0	-1
1	0	0	-1
1	0	1	1
1	1	0	0
1	1	1	2

$$b_2 = \neg(b_0 \wedge b_1) \iff b_0 + b_1 + 2b_2 - 2 \in \{0, 1\}$$

Proof for NAND-gate

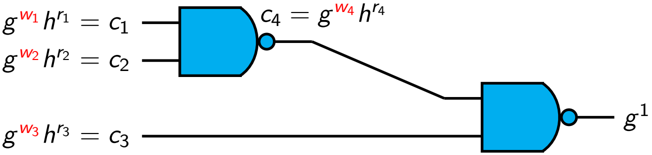


- Given c_1, c_2 and c_4 commitments for bits w_1, w_2, w_4
  $\rightsquigarrow$ Wish to prove $w_4 = \neg(w_1 \wedge w_2)$.
  i.e. $w_1 + w_2 + 2w_4 - 2 \in \{0, 1\}$
- We have

$$\begin{aligned} c_1 c_2 c_4^2 g^{-2} &= (g^{w_0} h^{r_0}) \cdot (g^{w_1} h^{r_1}) \cdot (g^{w_4} h^{r_4})^2 g^{-2} \\ &= g^{w_0 + w_1 + 2w_4 - 2} h^{r_0 + r_1 + 2r_4} \end{aligned}$$

- Prove that $c_1 c_2 c_4^2 g^{-2}$ contains 0 or 1

NIZK Proof for Circuit SAT



- Prove $w_i \in \{0, 1\}$ for $i \in \{1, 2, 3, 4\} \rightarrow 2k$ bits
  Prove $w_4 = \neg(w_1 \wedge w_2) \rightarrow k$ bits
  Prove $1 = \neg(w_3 \wedge w_4) \rightarrow k$ bits
- CRS size: $3k$ **bits**
  Proof size: $(2|W| + |C|)k$ **bits**

Groth-Ostrowsky-Sahai is ZK

Subgroup Membership Assumption

Hard to distinguish $h \in \mathbb{G}$ of order q from random h of order n

Simulation

- simulated CRS

h of order n by choosing $g = h^\tau$

- the simulation trapdoor is τ
- $\rightsquigarrow$ **perfectly hiding trapdoor commitments**

Groth-Ostrowsky-Sahai is ZK

Subgroup Membership Assumption

Hard to distinguish $h \in \mathbb{G}$ of order q from random h of order n

Simulation

- simulated CRS

h *of order* n *by choosing* $g = h^\tau$

- the simulation trapdoor is τ
- $\rightsquigarrow$ **perfectly hiding trapdoor commitments**

Groth-Ostrowsky-Sahai is ZK

Subgroup Membership Assumption

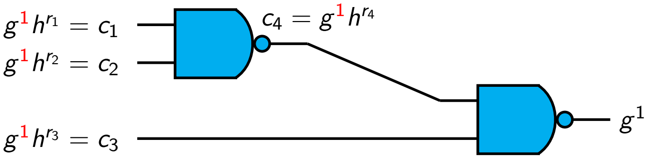
Hard to distinguish $h \in \mathbb{G}$ of order q from random h of order n

Simulation

- simulated CRS

h of order n by choosing $g = h^\tau$

- the simulation trapdoor is τ
- $\rightsquigarrow$ **perfectly hiding trapdoor commitments**



Groth-Ostrowsky-Sahai is ZK

Witness-indistinguishable 0/1-proof

- $c_1 = g^1 h^{r_1}$
  - $\pi_1 = (gh^{r_1})^{r_1}$ is the proof that c_1 contains 1
- $c_1 = g^1 h^{r_1} = g^0 g h^{r_1} = g^0 h^{\tau + r_1}$
  - $\pi_0 = (g^{-1} h^{\tau + r_1})^{\tau + r_1}$ is the proof that c_1 contains 0

$$\pi_0 = (g^{-1} h^{\tau + r_1})^{\tau + r_1} = (g^{-1} h^{\tau})^{\tau + r_1} (h^{r_1})^{r_1 + \tau} = (h^{r_1 + \tau})^{r_1} = (g^1 h^{r_1})^{r_1} = \pi_1$$

Witness-indistinguishable NAND-proof

- We have

$$\begin{aligned} c_1 c_2 c_4^2 g^{-2} &= (g^1 h^{r_1}) \cdot (g^1 h^{r_2}) \cdot (g^1 h^{r_4})^2 g^{-2} \\ &= g^2 h^{r_0 + r_1 + 2r_4} \\ &= g^1 h^{\tau + r_1 + r_2 + 2r_4} \end{aligned}$$

Computational ZK $\rightarrow$ Subgroup membership assumption

Groth-Ostrovsky-Sahai: Summary

- Perfect completeness and soundness, computational **zero-knowledge** for **NP**

- **Idea:**
  - Commit **bits** using BGN encryption
  - Prove the validity using homomorphic properties

Plug the commitments $\vec{c}$ in the equations and provide additionnal group element $\vec{\pi}$ to check the validity

$$e(g^w, g^w g^{-1}) = 1 \rightsquigarrow e(c, cg^{-1}) = e(h, \pi)$$

- Common reference string: $O(k)$ bits
- Proof: $\mathbf{O}(|\mathbf{C}|\mathbf{k})$ bits

Groth-~~Ostrovsky~~-Sahai: Summary

- Perfect completeness and soundness, computational **zero-knowledge** for **NP**

- **Idea:**
  - Commit **bits** using BGN encryption
  - Prove the validity using homomorphic properties

Plug the commitments $\vec{c}$ in the equations and provide additionnal group element $\vec{\pi}$ to check the validity

$$e(g^w, g^w g^{-1}) = 1 \rightsquigarrow e(c, cg^{-1}) = e(h, \pi)$$

- Common reference string: $O(k)$ bits
- Proof: $\mathbf{O}(|\mathbf{C}|\mathbf{k})$ bits

Groth-~~Ostrovsky~~-Sahai: Summary

witness-indistinguishability

- Perfect completeness and soundness, ~~computational **zero-knowledge**~~ for **NP**

- **Idea:**
  - Commit **bits** using BGN encryption
  - Prove the validity using homomorphic properties

Plug the commitments $\vec{c}$ in the equations and provide additionnal group element $\vec{\pi}$ to check the validity

$$e(g^w, g^w g^{-1}) = 1 \rightsquigarrow e(c, cg^{-1}) = e(h, \pi)$$

- Common reference string: $O(k)$ bits
- Proof: $\mathbf{O}(|\mathbf{C}|\mathbf{k})$ bits

Groth-~~Ostrovsky~~-Sahai: Summary

- Perfect completeness and soundness, ~~computational~~ ~~**zero-knowledge**~~ witness-indistinguishability for ~~**NP**~~ algebraic languages
- **Idea:**
  - Commit **bits** using BGN encryption
  - Prove the validity using homomorphic properties

> *Plug the commitments $\vec{c}$ in the equations and provide additionnal group element $\vec{\pi}$ to check the validity*

$$e(g^w, g^w g^{-1}) = 1 \rightsquigarrow e(c, cg^{-1}) = e(h, \pi)$$

- Common reference string: $O(k)$ bits
- Proof: $\mathbf{O}(|\mathbf{C}|\mathbf{k})$ bits

Groth-~~Ostrovsky~~-Sahai: Summary

witness-indistinguishability

- Perfect completeness and soundness, ~~computational **zero-knowledge**~~ for ~~**NP**~~ algebraic languages
- **Idea:** group elements
  - Commit ~~**bits**~~ using BGN encryption
  - Prove the validity using homomorphic properties

Plug the commitments $\vec{c}$ in the equations and provide additionnal group element $\vec{\pi}$ to check the validity

$$e(g^w, g^w g^{-1}) = 1 \rightsquigarrow e(c, cg^{-1}) = e(h, \pi)$$

- Common reference string: $O(k)$ bits
- Proof: $\mathbf{O}(|\mathbf{C}|\mathbf{k})$ bits

Groth-~~Ostrovsky~~-Sahai: Summary

witness-indistinguishability

- Perfect completeness and soundness, ~~computational **zero-knowledge**~~ for ~~**NP**~~ algebraic languages

- **Idea:** group elements
  - Commit ~~**bits**~~ using ~~BGN~~ encryption
  - Prove the validity using homomorphic properties

Plug the commitments $\vec{c}$ in the equations and provide additionnal group element $\vec{\pi}$ to check the validity

$$e(g^w, g^w g^{-1}) = 1 \rightsquigarrow e(c, cg^{-1}) = e(h, \pi)$$

- Common reference string: $O(k)$ bits
- Proof: $\mathbf{O}(|\mathbf{C}|\mathbf{k})$ bits

Groth-~~Ostrovsky~~-Sahai: Summary

witness-indistinguishability

- Perfect completeness and soundness, ~~computational~~ ~~**zero-knowledge**~~ for ~~**NP**~~ algebraic languages

- **Idea:** group elements
  - Commit ~~**bits**~~ using ~~BGN~~ encryption
  - Prove the validity using homomorphic properties

Plug the commitments $\vec{c}$ in the equations and provide additionnal group element $\vec{\pi}$ to check the validity

$$e(g^w, g^w g^{-1}) = 1 \rightsquigarrow e(c, cg^{-1}) = e(h, \pi)$$

- Common reference string: $O(k)$ bits
- Proof: ~~$\mathbf{O}(|\mathbf{C}|\mathbf{k})$~~ $O(|E|k)$ bits

Asymmetric bilinear structure

$(e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, p)$ bilinear structure:

- $\mathbb{G}_1$, $\mathbb{G}_2$ $\mathbb{G}_T$ multiplicative groups of order p
  - $p =$ **prime integer**
- $\langle g_i \rangle = \mathbb{G}_i$
- $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$
  - $\langle e(g_1, g_2) \rangle = \mathbb{G}_\mathbb{T}$
  - $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$, $a, b \in \mathbb{Z}$
- $\left.\begin{array}{l}\text{deciding group membership,}\\ \text{group operations,}\\ \text{bilinear map}\end{array}\right\}$ efficiently computable.

ElGamal Encryption Scheme

Public key: $(e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, p)$
$g_i, u_i = g_i^x \in \mathbb{G}$

Secret key: x

Encryption: $(c_1, c_2) = (g_1^\alpha, mu_i^{\alpha+\beta})\ (\alpha, \beta \xleftarrow{R} \mathbb{Z}_p)$

Decryption: $c_2/(c_1^x = m$

IND-CPA-secure under the:

Decision Diffie-Hellman Assumption in $\mathbb{G}_i$

given (g_i, h_i, g_i^α), Hard to distinguish h_i^α from random

Double ElGamal Commitment Scheme

Commitment key: $(e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, p)$
$u \in \mathbb{G}_1^{2\times 2}$,
$v \in \mathbb{G}_2^{2\times 2}$

Commitment in $\mathbb{G}_a$: $(c_1, c_2) = (u_{1,1}^{\alpha} u_{2,1}^{\beta}, m u_{1,2}^{\alpha} u_{2,2}^{\beta})$

- **Perfectly binding:** if $u = (u_{1,1} = g, u_{1,2} = g^{\mu}, u_{2,1} = g^{\nu}, u_{2,2} = g^{\mu\nu})$
- **Perfectly hiding:** if $u = (u_{1,1} = g, u_{1,2} = g^{\mu}, u_{2,1} = g^{\nu}, u_{2,2} = g^{\mu\nu+1})$
- **Homomorphic:** $(c_1, c_2) \cdot (c_1', c_2') = (u_{1,1}^{\alpha+\alpha'} u_{2,1}^{\beta+\beta'}, (mm') u_{1,2}^{\alpha+\alpha'} u_{2,2}^{\beta+\beta'})$

Keys are indistinguishable under DDH Assumption in $\mathbb{G}_1$ **and** $\mathbb{G}_2$ $\rightsquigarrow$ SXDH

Groth-Sahai Proof System

Groth-Sahai Proof System

- **Pairing product equation (PPE):** for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}_1$, $\mathcal{Y}_1, \ldots, \mathcal{Y}_m \in \mathbb{G}_2$

$$(E) : \prod_{i=1}^{n} e(\mathcal{X}_i, A_i) \prod_{j=1}^{m} e(B_j, \mathcal{Y}_j) \prod_{i=1}^{n} \prod_{j=1}^{m} e(\mathcal{X}_i, \mathcal{Y}_j)^{\gamma_{i,j}} = t_T$$

  determined by $A_i \in \mathbb{G}_2$, $B_j \in \mathbb{G}_1$, $\gamma_{i,j} \in \mathbb{Z}_p$ and $t_T \in \mathbb{G}_T$.

- Groth-Sahai $\rightsquigarrow$ WI proofs that elements in $\mathbb{G}$ that were committed to satisfy PPE

Assumption	SXDH	SD
Variables $\in \mathbb{G}$	2	1
PPE	(4,4)	1
(Linear)	2	1
Verification	$5\,m + 3\,n + 16$ P	$n + 1$ P

O. Blazy, G. Fuchsbauer, M. Izabachène, A. Jambert, H. Sibert, D. V.
Batch Groth-Sahai.
ACNS 2010

Groth-Sahai Proof System

Groth-Sahai Proof System

- **Pairing product equation (PPE):** for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}_1$, $\mathcal{Y}_1, \ldots, \mathcal{Y}_m \in \mathbb{G}_2$

$$(E): \prod_{i=1}^{n} e(\mathcal{X}_i, A_i) \prod_{j=1}^{m} e(B_j, \mathcal{Y}_j) \prod_{i=1}^{n} \prod_{j=1}^{m} e(\mathcal{X}_i, \mathcal{Y}_j)^{\gamma_{i,j}} = t_T$$

  determined by $A_i \in \mathbb{G}_2$, $B_j \in \mathbb{G}_1$, $\gamma_{i,j} \in \mathbb{Z}_p$ and $t_T \in \mathbb{G}_T$.

- Groth-Sahai $\rightsquigarrow$ WI proofs that elements in $\mathbb{G}$ that were committed to satisfy PPE

Assumption	SXDH	SD
Variables $\in \mathbb{G}$	2	1
PPE	(4,4)	1
(Linear)	2	1
Verification	$5\,m + 3\,n + 16$ P	$n + 1$ P

O. Blazy, G. Fuchsbauer, M. Izabachène, A. Jambert, H. Sibert, D. V.
Batch Groth-Sahai.
ACNS 2010

Groth-Sahai Proof System

Groth-Sahai Proof System

- **Pairing product equation (PPE):** for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}_1$, $\mathcal{Y}_1, \ldots, \mathcal{Y}_m \in \mathbb{G}_2$

$$(E): \prod_{i=1}^{n} e(\mathcal{X}_i, A_i) \prod_{j=1}^{m} e(B_j, \mathcal{Y}_j) \prod_{i=1}^{n} \prod_{j=1}^{m} e(\mathcal{X}_i, \mathcal{Y}_j)^{\gamma_{i,j}} = t_T$$

  determined by $A_i \in \mathbb{G}_2$, $B_j \in \mathbb{G}_1$, $\gamma_{i,j} \in \mathbb{Z}_p$ and $t_T \in \mathbb{G}_T$.

- Groth-Sahai $\rightsquigarrow$ WI proofs that elements in $\mathbb{G}$ that were committed to satisfy PPE

Assumption	SXDH	SD
Variables $\in \mathbb{G}$	2	1
PPE	(4,4)	1
(Linear)	2	1
Verification	$m + 2n + 8$ P	$n + 1$ P

O. Blazy, G. Fuchsbauer, M. Izabachène, A. Jambert, H. Sibert, D. V.
Batch Groth-Sahai.
ACNS 2010

Groth-Sahai Proof System: NIWI

$$(E) : \prod_{i=1}^{n} e(\mathcal{X}_i, A_i) \prod_{j=1}^{m} e(B_j, \mathcal{Y}_j) \prod_{i=1}^{n} \prod_{j=1}^{m} e(\mathcal{X}_i, \mathcal{Y}_j)^{\gamma_{i,j}} = t_T$$

- Setup: on input the bilinear group $\rightsquigarrow$ output a commitment key **ck**
- Com: on input **ck**, $X \in \mathbb{G}$, randomness $\rho \rightsquigarrow$ output commitment $\vec{c_X}$ to X
- Prove: on input **ck**, $(X_i, \rho_i)_{i=1,\dots,n}$ and $(E) \rightsquigarrow$ output a proof ϕ
- Verify: on input **ck**, $\vec{c_{X_i}}$, (E) and $\phi \rightsquigarrow$ output 0 or 1

Properties:

- correctness: honestly generated proofs are accepted by **Verify**
- soundness: perfectly binding key
- witness-indistinguishability: perfectly hiding key

Remark: such equations are not known to always have NIZK proofs

Groth-Sahai Proof System: NIWI

$$(E): \prod_{i=1}^{n} e(\mathcal{X}_i, A_i) \prod_{j=1}^{m} e(B_j, \mathcal{Y}_j) \prod_{i=1}^{n}\prod_{j=1}^{m} e(\mathcal{X}_i, \mathcal{Y}_j)^{\gamma_{i,j}} = t_T$$

Setup on input the bilinear group $\rightsquigarrow$ output a commitment key **ck**

Com on input **ck**, $X \in \mathbb{G}$, randomness $\rho \rightsquigarrow$ output commitment $\vec{c_X}$ to X

Prove on input **ck**, $(X_i, \rho_i)_{i=1,\ldots,n}$ and $(E) \rightsquigarrow$ output a proof ϕ

Verify on input **ck**, $\vec{c_{X_i}}$, (E) and $\phi \rightsquigarrow$ output 0 or 1

Properties:

- correctness: honestly generated proofs are accepted by **Verify**
- soundness: perfectly binding key
- witness-indistinguishability: perfectly hiding key

Remark: such equations are not known to always have NIZK proofs

Groth-Sahai Proof System: NIWI

$$(E) : \prod_{i=1}^{n} e(\mathcal{X}_i, A_i) \prod_{j=1}^{m} e(B_j, \mathcal{Y}_j) \prod_{i=1}^{n} \prod_{j=1}^{m} e(\mathcal{X}_i, \mathcal{Y}_j)^{\gamma_{i,j}} = t_T$$

Setup on input the bilinear group $\rightsquigarrow$ output a commitment key **ck**

Com on input **ck**, $X \in \mathbb{G}$, randomness $\rho \rightsquigarrow$ output commitment $\vec{c_X}$ to X

Prove on input **ck**, $(X_i, \rho_i)_{i=1,\ldots,n}$ and $(E) \rightsquigarrow$ output a proof ϕ

Verify on input **ck**, $\vec{c_{X_i}}$, (E) and $\phi \rightsquigarrow$ output 0 or 1

Properties:

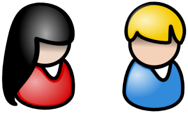
- correctness: honestly generated proofs are accepted by **Verify**
- soundness: perfectly binding key
- witness-indistinguishability: perfectly hiding key

Remark: such equations are not known to always have NIZK proofs

Contents

Transferable Fair E-cash: Cast of characters

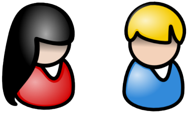
Users



Alice Bob

Users: withdraw, transfer or spend coins
(registered to a system manager $\mathcal{S}$)

Transferable Fair E-cash: Cast of characters

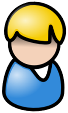


Shop: to which coins are spent

Transferable Fair E-cash: Cast of characters

Users

Alice



Bob

Shop

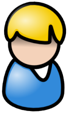
Bank

Bank $\mathcal{B}$: issue coins

Transferable Fair E-cash: Cast of characters

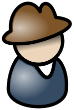

Users


Alice


Bob


Shop


Bank

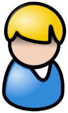

Double-spending detector

Double-spending detector $\mathcal{D}$: check (on deposit) if a coin has already been spent (coins can be easily duplicated $\rightsquigarrow$ copies of cash should not be spendable.)

Transferable Fair E-cash: Cast of characters

Users

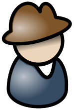
Alice



Bob

Shop

Bank



Double-spending detector

Tracer

Tracer $\mathcal{T}$: trace coins, revoke anonymity and identify double-spenders.

Transferable E-cash: Our Construction

- in our scheme, coins are transferable while remaining constant in size
- we circumvent the impossibility with a new method to trace double spenders:
  - users keep **receipts** when receiving coins
    (instead of storing all information about transfers inside the coin)
- anonymous w.r.t. an entity issuing coins **and** able to detect double spendings.
- the construction: our new primitive + the Groth-Sahai proof system

G. Fuchsbauer, D. Pointcheval, D. V.
Transferable Constant-Size Fair E-Cash.
CANS 2009

A New Primitive: Partially-Blind Certification

= 4-tuple of (interactive) PPTs:

- **Setup**: $k \rightsquigarrow (pk, sk)$
- **Sign** and **User** are interactive PPTs s.t.:
  - **User**: $pk \rightsquigarrow (\sigma, \tau)$ or $\perp$
  - **Sign**: $sk \rightsquigarrow$ `completed` or `not-completed`

  (certificate issuing protocol)
- **Verif**: $(pk, (\sigma, \tau)) \rightsquigarrow$ `accept` or `reject`.

1. (σ, τ) = certificate for pk
2. τ = blind component of the certificate.
3. **Properties:**
   - correctness
   - partial blindness: τ is only known to the user and cannot be associated to a particular protocol execution by the issuer
   - unforgeability: from m runs of the protocol, it is impossible to derive more than m valid certificates

A New Primitive: Partially-Blind Certification

= 4-tuple of (interactive) PPTs:

- **Setup**: $k \rightsquigarrow (pk, sk)$
- **Sign** and **User** are interactive PPTs s.t.:
  - **User**: $pk \rightsquigarrow (\sigma, \tau)$ or $\perp$
  - **Sign**: $sk \rightsquigarrow$ `completed` or `not-completed`

  (certificate issuing protocol)
- **Verif**: $(pk, (\sigma, \tau)) \rightsquigarrow$ `accept` or `reject`.

1. $(\sigma, \tau) =$ certificate for pk
2. $\tau =$ blind component of the certificate.
3. **Properties:**
   - correctness
   - partial blindness: τ is only known to the user and cannot be associated to a particular protocol execution by the issuer
   - unforgeability: from m runs of the protocol, it is impossible to derive more than m valid certificates

Partially-Blind Certification: Instantiation

(1) User Choose $r, y_1 \leftarrow \mathbb{Z}_p$, compute and send: $R_1 := (g_1^{y_1} h_1)^r, \quad T := g_1^r$
and zero-knowledge proofs of knowledge of r and y_1

(2) Signer Choose $s, y_2 \leftarrow \mathbb{Z}_p$ and compute $R := R_1 T^{y_2}$
(note that $R = (h_1 g_1^y)^r$ with $y := y_1 + y_2$.)

Send
$\left(S_1 := R^{\frac{1}{x+s}},\ S_2 := g_1^s,\ S_3 := g_2^s,\ S_4 := g_1^{y_2},\ S_5 := g_2^{y_2}\right)$

(3) User Check whether $(S_1, S_2, S_3, S_4, S_5)$ is correctly formed:

$$e(S_2, g_2) \stackrel{?}{=} e(g_1, S_3) \quad e(S_4, g_2) \stackrel{?}{=} e(g_1, S_5) \quad e(S_1, XS_2) \stackrel{?}{=} e(R, g_2)$$

If so, compute a certificate

$$\boxed{\left(C_1 := S_1^{1/r},\ C_2 := S_2,\ C_3 := S_3,\ C_4 := g_1^{y_1} S_4 = g_1^y,\ C_5 := g_2^{y_1} S_5 = g_2^y\right)}$$

Transferable Constant-Size Fair E-Cash

- the core of a coin in our system is a partially-blind certificate.
- **Withdrawal:** partially blind issuing ⇝ the bank does not know C_5.
- **Spend/Transfer**: the user commit to the coin and prove validity. Transfer ⇝ re-randomize the encryption ⇝ unlinkable anonymity.
- **Double-spending detection:** the detector has the decryption key to compare encrypted certificates.
  - ⇝ does not guarantee user anonymity when bank and detector cooperate.
  - C_5 is thus encrypted under a different key than the rest
  - the detector gets only the key to decrypt C_5, which suffices to detect double spending.
- **Traceability:** the receipts, given when transferring coins, are group signatures on them
- **Double-spender identification:** the tracer follows backwards the paths the certificate took before reaching the spender, by opening the receipts. A user that spent or transferred a coin twice is then unable to show two receipts.

Transferable Constant-Size Fair E-Cash

- the core of a coin in our system is a partially-blind certificate.
- **Withdrawal:** partially blind issuing $\rightsquigarrow$ the bank does not know C_5.
- **Spend/Transfer**: the user commit to the coin and prove validity. Transfer $\rightsquigarrow$ re-randomize the encryption $\rightsquigarrow$ unlinkable anonymity.
- **Double-spending detection:** the detector has the decryption key to compare encrypted certificates.
  - $\rightsquigarrow$ does not guarantee user anonymity when bank and detector cooperate.
  - C_5 is thus encrypted under a different key than the rest
  - the detector gets only the key to decrypt C_5, which suffices to detect double spending.
- **Traceability:** the receipts, given when transferring coins, are group signatures on them
- **Double-spender identification:** the tracer follows backwards the paths the certificate took before reaching the spender, by opening the receipts. A user that spent or transferred a coin twice is then unable to show two receipts.

Transferable Constant-Size Fair E-Cash

- the core of a coin in our system is a partially-blind certificate.
- **Withdrawal:** partially blind issuing $\rightsquigarrow$ the bank does not know C_5.
- **Spend/Transfer**: the user commit to the coin and prove validity. Transfer $\rightsquigarrow$ re-randomize the encryption $\rightsquigarrow$ unlinkable anonymity.
- **Double-spending detection:** the detector has the decryption key to compare encrypted certificates.
  - $\rightsquigarrow$ does not guarantee user anonymity when bank and detector cooperate.
  - C_5 is thus encrypted under a different key than the rest
  - the detector gets only the key to decrypt C_5, which suffices to detect double spending.
- **Traceability:** the receipts, given when transferring coins, are group signatures on them
- **Double-spender identification:** the tracer follows backwards the paths the certificate took before reaching the spender, by opening the receipts. A user that spent or transferred a coin twice is then unable to show two receipts.

Transferable Constant-Size Fair E-Cash

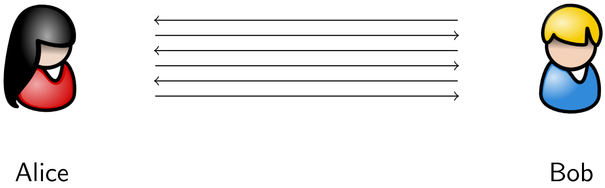
- the core of a coin in our system is a partially-blind certificate.
- **Withdrawal:** partially blind issuing $\rightsquigarrow$ the bank does not know C_5.
- **Spend/Transfer**: the user commit to the coin and prove validity. Transfer $\rightsquigarrow$ re-randomize the encryption $\rightsquigarrow$ unlinkable anonymity.
- **Double-spending detection:** the detector has the decryption key to compare encrypted certificates.
  - $\rightsquigarrow$ does not guarantee user anonymity when bank and detector cooperate.
  - C_5 is thus encrypted under a different key than the rest
  - the detector gets only the key to decrypt C_5, which suffices to detect double spending.
- **Traceability:** the receipts, given when transferring coins, are group signatures on them
- **Double-spender identification:** the tracer follows backwards the paths the certificate took before reaching the spender, by opening the receipts. A user that spent or transferred a coin twice is then unable to show two receipts.

Transferable Constant-Size Fair E-Cash

- the core of a coin in our system is a partially-blind certificate.
- **Withdrawal:** partially blind issuing $\rightsquigarrow$ the bank does not know C_5.
- **Spend/Transfer**: the user commit to the coin and prove validity. Transfer $\rightsquigarrow$ re-randomize the encryption $\rightsquigarrow$ unlinkable anonymity.
- **Double-spending detection:** the detector has the decryption key to compare encrypted certificates.
  - $\rightsquigarrow$ does not guarantee user anonymity when bank and detector cooperate.
  - C_5 is thus encrypted under a different key than the rest
  - the detector gets only the key to decrypt C_5, which suffices to detect double spending.
- **Traceability:** the receipts, given when transferring coins, are group signatures on them
- **Double-spender identification:** the tracer follows backwards the paths the certificate took before reaching the spender, by opening the receipts. A user that spent or transferred a coin twice is then unable to show two receipts.

Contents

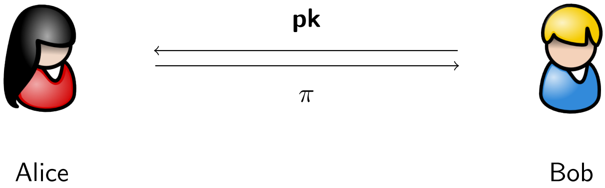
Zero-knowledge Interactive Proof



- interactive method for one party to prove to another that a statement $\mathcal{S}$ is true, without revealing anything other than the veracity of $\mathcal{S}$.

1. **Completeness:** $\mathcal{S}$ is true $\rightsquigarrow$ verifier will be convinced of this fact
2. **Soundness:** $\mathcal{S}$ is false $\rightsquigarrow$ no cheating prover can convince the verifier that $\mathcal{S}$ is true
3. **Zero-knowledge:** $\mathcal{S}$ is true $\rightsquigarrow$ no cheating verifier learns anything other than this fact.

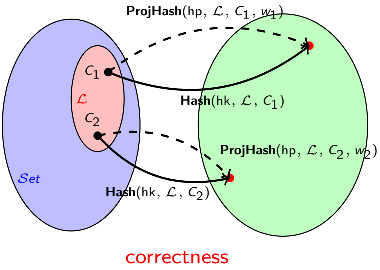
Designated Verifier Zero-Knowledge Proofs



- interactive method for one party to prove to another that a statement $\mathcal{S}$ is true, without revealing anything other than the veracity of $\mathcal{S}$.

1. **Completeness:** $\mathcal{S}$ is true $\rightsquigarrow$ verifier will be convinced of this fact
2. **Soundness:** $\mathcal{S}$ is false $\rightsquigarrow$ no cheating prover can convince the verifier that $\mathcal{S}$ is true
3. **Zero-knowledge:** $\mathcal{S}$ is true $\rightsquigarrow$ no cheating verifier learns anything other than this fact.

Smooth-Projective Hash Functions



- **HashKG**($\mathcal{L}$) generates a hashing key hk for the language $\mathcal{L}$;
- **ProjKG**(hk, $\mathcal{L}$, C) derives the projection key hp, possibly depending on a word $C \in \mathcal{S}et$;
- **Hash**(hk, $\mathcal{L}$, C) outputs the hash value of the word C from the hashing key;
- **ProjHash**(hp, $\mathcal{L}$, C, w) outputs the hash value of the word C from the projection key hp, and the witness w that $C \in \mathcal{L}$.

Proof of a Diffie Hellman tuple

Given a group $\mathbb{G}$ of order p, with a generators g_1 and g_2

$$\mathcal{L} = \{(g_1^r, g_2^r), r \in \mathbb{Z}_p^*\} \subset \mathbb{G}^2 = \mathcal{S}et$$

(Cramer-Shoup) SPHF:

- **HashKG**($\mathcal{L}$) generates a hashing key $\mathsf{hk} = (x_1, x_2) \stackrel{\$}{\leftarrow} \mathbb{Z}_p^2$;
- **ProjKG**($\mathsf{hk}, \mathcal{L}, \perp$) derives the projection key $\mathsf{hp} = g_1^{x_1} g_2^{x_2}$.
- **Hash**($\mathsf{hk}, \mathcal{L}, C = (u_1, u_2)$) outputs the hash value $H = u_1^{x_1} \cdot u_2^{x_2} \in \mathbb{G}$.
- **ProjHash**($\mathsf{hp}, \mathcal{L}, C = (g_1^r, g_2^r), w = r$) outputs the hash value $H' = \mathsf{hp}^r \in G$.

Proof of a Diffie Hellman tuple

Given a group $\mathbb{G}$ of order p, with a generators g_1 and g_2

$$\mathcal{L} = \{(g_1^r, g_2^r), r \in \mathbb{Z}_p^*\} \subset \mathbb{G}^2 = \mathcal{S}et$$

(Cramer-Shoup) SPHF:

- **HashKG**($\mathcal{L}$) generates a hashing key $\mathsf{hk} = (x_1, x_2) \stackrel{\$}{\leftarrow} \mathbb{Z}_p^2$;
- **ProjKG**($\mathsf{hk}, \mathcal{L}, \perp$) derives the projection key $\mathsf{hp} = g_1^{x_1} g_2^{x_2}$.
- **Hash**($\mathsf{hk}, \mathcal{L}, C = (u_1, u_2)$) outputs the hash value $H = u_1^{x_1} \cdot u_2^{x_2} \in \mathbb{G}$.
- **ProjHash**($\mathsf{hp}, \mathcal{L}, C = (g_1^r, g_2^r), w = r$) outputs the hash value $H' = \mathsf{hp}^r \in G$.

Proof of the Encryption of One Bit

Given a group $\mathbb{G}$ of order p, with a generators g_1, g_2 and u

$$\mathcal{L} = \{C = (c_1, c_2) \in \mathbb{G}^2, \exists r \in \mathbb{Z}_p, c_1 = g_1^r \wedge c_2 \in \{g_2^r, g_2^r \cdot u\}\} \subset \mathbb{G}^2 = \mathcal{S}et$$

(Benhamouda, Blazy, Chevalier, Pointcheval, V.) SPHF:

- **HashKG**($\mathcal{L}$): $\mathsf{hk} = ((x_1, x_2), (y_1, y_2)) \xleftarrow{\$} \mathbb{Z}_p^4$
- **ProjKG**($\mathsf{hk}, \mathcal{L}, C$): $\mathsf{hp} = (g_1^{x_1} g_2^{x_2}, g_1^{y_1} g_2^{y_2}, \mathsf{hp}_\Delta = c_1^{x_1} c_2^{x_2} \cdot c_1^{y_1} (c_2/u)^{y_2})$
- **Hash**($\mathsf{hk}, \mathcal{L}, C$): $v = c_1^{x_1} c_2^{x_2}$
- **ProjHash**($\mathsf{hp}, \mathcal{L}, C, r$): If $c_2 = g_2^r$, $v' = \mathsf{hp}_1^r$,

  else (if $c_2 = g_2^r \cdot u$), $v' = \mathsf{hp}_\Delta / \mathsf{hp}_2^r$

Application: $\rightsquigarrow$ efficient blind signatures (w/o random oracles)

Proof of the Encryption of One Bit

Given a group $\mathbb{G}$ of order p, with a generators g_1, g_2 and u

$$\mathcal{L} = \{C = (c_1, c_2) \in \mathbb{G}^2, \exists r \in \mathbb{Z}_p, c_1 = g_1^r \wedge c_2 \in \{g_2^r, g_2^r \cdot u\}\} \subset \mathbb{G}^2 = \mathcal{S}et$$

(Benhamouda, Blazy, Chevalier, Pointcheval, V.) SPHF:

- **HashKG**($\mathcal{L}$): $\mathsf{hk} = ((x_1, x_2), (y_1, y_2)) \stackrel{\$}{\leftarrow} \mathbb{Z}_p^4$
- **ProjKG**($\mathsf{hk}, \mathcal{L}, C$): $\mathsf{hp} = (g_1^{x_1} g_2^{x_2}, g_1^{y_1} g_2^{y_2}, \mathsf{hp}_\Delta = c_1^{x_1} c_2^{x_2} \cdot c_1^{y_1} (c_2/u)^{y_2})$
- **Hash**($\mathsf{hk}, \mathcal{L}, C$): $v = c_1^{x_1} c_2^{x_2}$
- **ProjHash**($\mathsf{hp}, \mathcal{L}, C, r$): If $c_2 = g_2^r$, $v' = \mathsf{hp}_1^r$,

  else (if $c_2 = g_2^r \cdot u$), $v' = \mathsf{hp}_\Delta / \mathsf{hp}_2^r$

Application: $\rightsquigarrow$ efficient blind signatures (w/o random oracles)

Other Applications . . .

O. Blazy, D. Pointcheval, D. V.
Round-Optimal Privacy-Preserving Protocols with Smooth Projective Hash Functions
TCC 2012

F. Benhamouda, O. Blazy, C. Chevalier, D. Pointcheval, D. V.
Efficient UC-Secure Authenticated Key-Exchange for Algebraic Languages
PKC 2013

O. Blazy, C. Chevalier, D. Pointcheval, D. V.
Analysis and Improvement of Lindell's UC-Secure Commitment Schemes
ACNS 2013

F. Benhamouda, O. Blazy, C. Chevalier, D. Pointcheval, D. V.
New Techniques for SPHFs and Efficient One-Round PAKE Protocols
Crypto 2013

Contents

Conclusion

- Groth-Sahai framework for NIWI/NIZK proofs
- (Smooth-Projective Hash Functions)
- **Applications**
  - group signatures, blind signatures, PAKE, ...
  - Efficient (offline) e-cash, e-voting systems, ...
- **Perspectives**
  - improve the efficiency of resulting protocols
    (recent advances in Groth-Sahai proofs/SPHF)
  - design tools for automatic generation Groth-Sahai proofs/SPHF