Signatures on Randomizable Ciphertexts

O. Blazy, G. Fuchsbauer, D. Pointcheval, D. Vergnaud

ENS / CNRS / INRIA

PKC 2011

1. Brief Overview
2. Cryptographic Tools
3. Our Instantiation
4. Applications

For dessert, we let people vote

- ✓ Chocolate Cake
- ✓ Cheese Cake
- ✓ Fruit Salad
- ✓ Brussels Sprout

After collection, we count the number of ballots:

Chocolate Cake	123
Cheese Cake	79
Fruit Salad	42
Brussels Sprout	1

Authentication

- Only people authorized to vote should be able to vote
- People should be able to vote only once

Anonymity

- Votes should be anonymous
- △ Receipt freeness: Prevent Vote Selling

Homomorphic Encryption and Signature approach

- The voter generates his vote v.
- The voter encrypts v to the server in c.
- The voter signs c and outputs σ.
- (c, σ) is a ballot unique per voter, and anonymous
- Counting: granted homomorphic encryption $C = \prod c$.
- The server decrypts C.

Vote v

$$\xrightarrow{c(v)}$$

$$\xrightarrow{\sigma(c;s)}$$

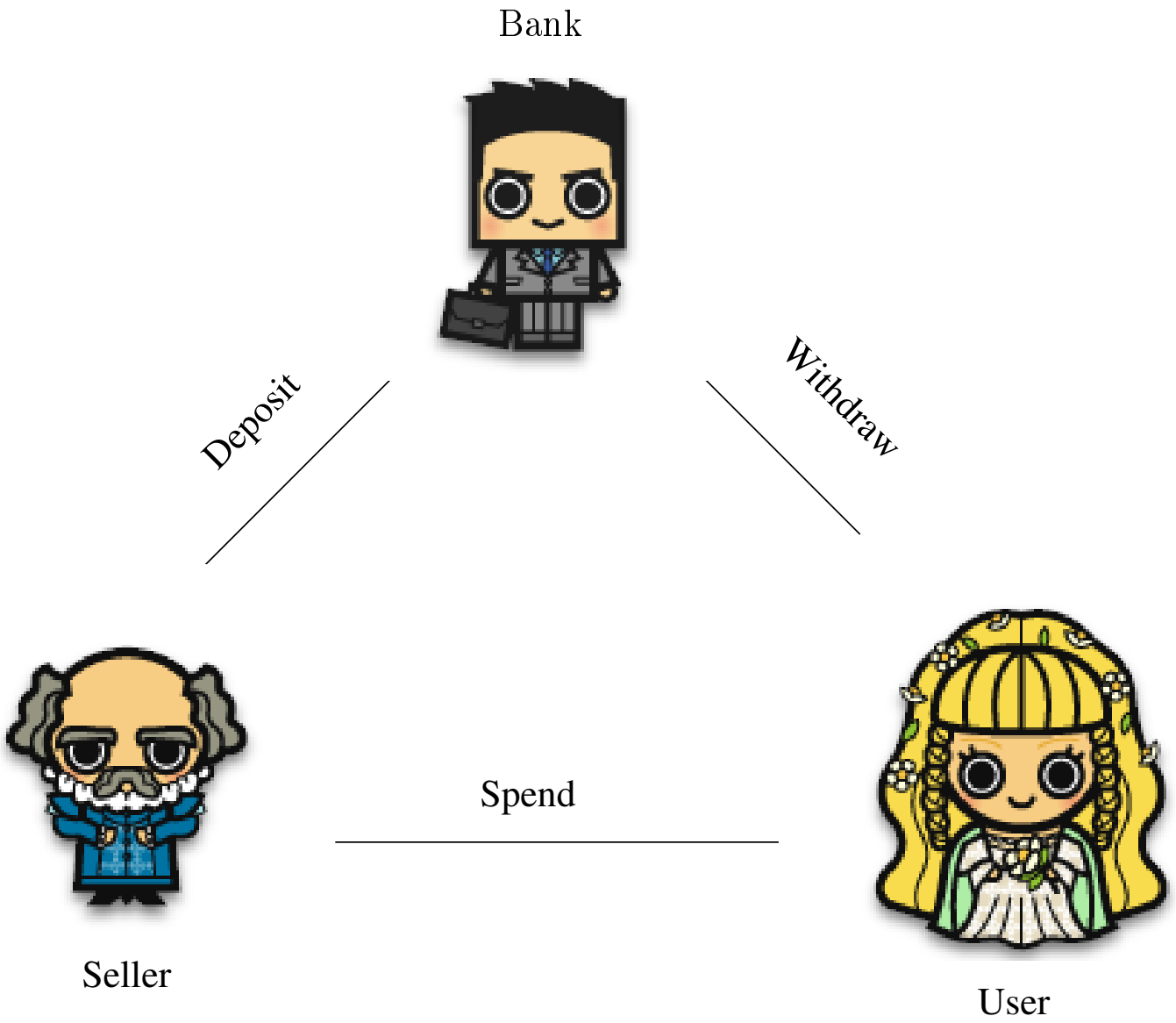
Vote v

$$\xrightarrow{\quad c(v) \quad}$$

$$\rightleftharpoons \xleftarrow{\quad c' \equiv c \quad} \rightleftharpoons$$

$$\xrightarrow{\quad \sigma'(c'; s') \quad}$$

E-Cash



Anonymity

- The bank cannot link a withdrawal to a deposit

No double-spending

- A coin should not be used twice

Definition (Blind Signature)

A blind signature allows a user to get a message m signed by an authority into σ so that it cannot recognize later the pair (m, σ).

Vote

- A user should be able to encrypt a ballot.
- He should be able to sign this encryption.
- Receiving this vote, one should be able to randomize for *Receipt-Freeness*.

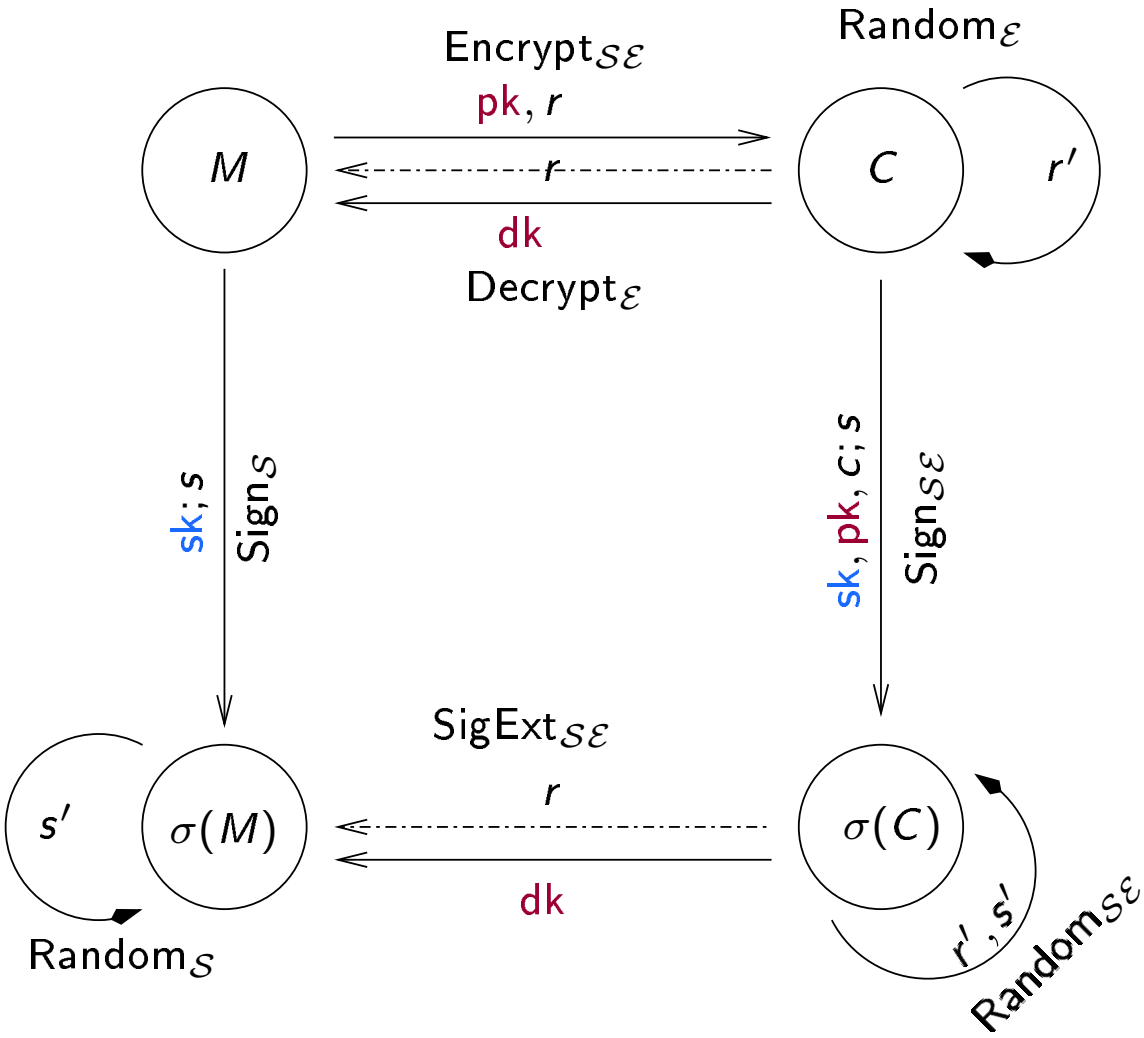
E-Cash

- A user should be able to encrypt a token
- The bank should be able to sign it providing *Unforgeability*
- This signature should now be able to be randomized to provide *Anonymity*

Our Solution

- Same underlying requirements;
- Advance security notions in both schemes requires to extract some kind of signature on the associated plaintext;
- General Framework for Signature on Randomizable Ciphertexts.
- $\rightsquigarrow$ Commutative encryption / signature.

Extractable Randomizable Signature on Ciphertexts



Definition (CDH)

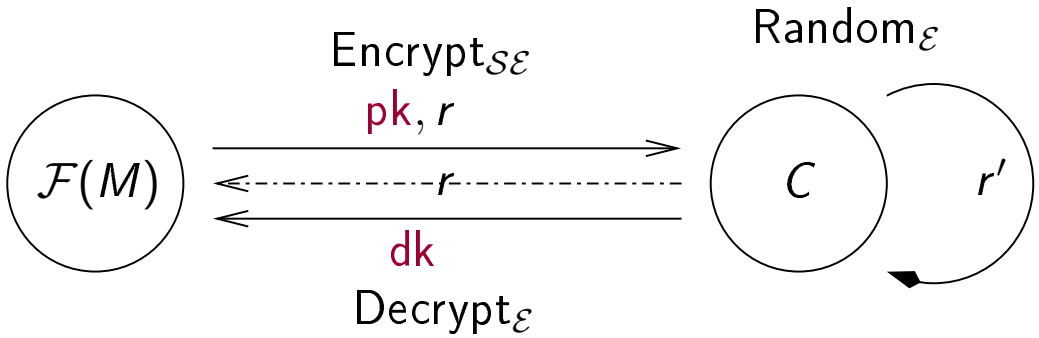
Given $g, g^a, h \in \mathbb{G}^3$, it is hard to compute h^a.

Definition (DLin)

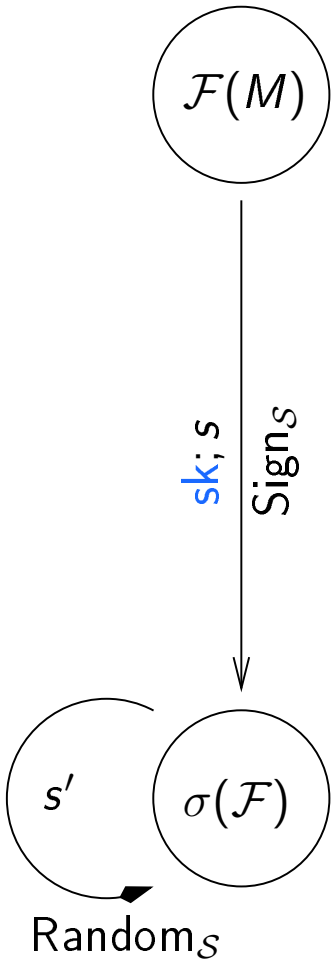
Given $u, v, w, u^a, v^b, w^c \in \mathbb{G}^6$, it is hard to decide whether $c = a + b$.

Definition (Encryption Scheme: Linear Encryption)

- Setup(1^k): $\mathbb{G}, w \in \mathbb{G}$;
- EKeyGen: $\mathsf{pk} = (u = w^\alpha, v = w^\beta)$, $\mathsf{dk} = (\alpha, \beta)$;
- Encrypt($\mathsf{pk}, X; r$): $C = (u^{r_1}, v^{r_2}, X \cdot w^{r_1 + r_2})$;
- Decrypt(dk, c): $X = C_3 / (C_1^{1/\alpha} \cdot C_2^{1/\beta})$.



Unlinkability on the randomization.



Definition (Signature Scheme: Waters Signature)

- Setup(1^k): $\mathbb{G}, g, h \in \mathbb{G}, \mathcal{F}$;
- SKeyGen: $\mathsf{vk} = g^x$, $\mathsf{sk} = h^x$;
- Sign$(\mathsf{sk}, m; s)$: $\sigma = (\mathsf{sk} \cdot \mathcal{F}(m)^s, g^s)$;
- Verif(vk, m, σ): $e(\sigma_1, g) = e(\mathcal{F}(m), \sigma_2) \cdot e(h, \mathsf{vk})$.

s can be randomized.

Definition (Groth Sahai Commitments)

Setup($\mathbb{G}$): $\mathbf{ck} = \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$

Com($\mathbf{ck}$, X; s): $(u_{1,1}^{s_1} \cdot u_{3,1}^{s_3}, u_{2,2}^{s_2} \cdot u_{3,2}^{s_3}, X \cdot g^{s_1+s_2} \cdot u_{3,3}^{s_3})$;

Prove($\mathbf{ck}$, (X, s), (E)): proof ϕ;

Verify($\mathbf{ck}$, $\vec{c_X}$, (E), ϕ): checks whether ϕ is valid.

Properties:

- soundness
- witness-indistinguishability
- Randomizability:

  Commitments and proofs are publicly randomizable and unlinkable.

Definition (Groth Sahai Commitments)

Setup($\mathbb{G}$): $\mathbf{ck} = \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$

Com($\mathbf{ck}$, X; s): $(u_{1,1}^{s_1} \cdot u_{3,1}^{s_3}, u_{2,2}^{s_2} \cdot u_{3,2}^{s_3}, X \cdot g^{s_1+s_2} \cdot u_{3,3}^{s_3})$;

Prove($\mathbf{ck}$, (X, s), (E)): proof ϕ;

Verify($\mathbf{ck}$, $\vec{c_X}$, (E), ϕ): checks whether ϕ is valid.

Properties:

- soundness
- witness-indistinguishability
- Randomizability:

  Commitments and proofs are publicly randomizable and unlinkable.

Definition (Groth Sahai Commitments)

Setup($\mathbb{G}$): $\mathbf{ck} = \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$

Com($\mathbf{ck}$, X; s): $(u_{1,1}^{s_1} \cdot u_{3,1}^{s_3}, u_{2,2}^{s_2} \cdot u_{3,2}^{s_3}, X \cdot g^{s_1+s_2} \cdot u_{3,3}^{s_3})$;

Prove($\mathbf{ck}$, (X, s), (E)): proof ϕ;

Verify($\mathbf{ck}$, $\vec{c_X}$, (E), ϕ): checks whether ϕ is valid.

Properties:

- soundness
- witness-indistinguishability
- Randomizability:

  Commitments and proofs are publicly randomizable and unlinkable.

Definition (Signature Scheme:Revisited Waters Signature)

- Setup(1^k): $\mathbb{G}, g, h \in \mathbb{G}$;
- SKeyGen: $\mathsf{vk} = g^x$, $\mathsf{sk} = h^x$;
- Sign$(\mathsf{sk}, m, R_1, R_2, T; s)$: On a valid tuple: $(e(R_1 R_2, \mathsf{vk}) = e(g, T))$: $(\sigma = (\mathsf{sk}(\mathcal{F}(m) R_1 R_2)^s, R_1^s, R_2^s, g^s)$;
- Verif(vk, m, σ) : $e(\sigma_1, g) = e(\mathcal{F}(m) R_1 R_2, \sigma_2).e(h, \mathsf{vk})$, $e(\sigma_3, g) = e(R_1, \sigma_5), e(\sigma_4, g) = e(R_2, \sigma_5)$.

Avoid some bit per bit proofs.
Same security assumption as standard Waters Signature.

Commutative properties

Encrypt

To encrypt a message m:

$$c = (\mathsf{pk}_1{}^{r_1}, \mathsf{pk}_2{}^{r_2}, \mathcal{F}(m) \cdot g^{r_1+r_2})$$

Commutative properties

Encrypt

To encrypt a message m:

$$c = (\mathsf{pk}_1{}^{r_1}, \mathsf{pk}_2{}^{r_2}, \mathcal{F}(m) \cdot g^{r_1+r_2})$$

Sign $\circ$ Encrypt

To sign a valid ciphertext c_1, c_2, c_3, one has simply to produce.

$$\sigma = (c_1{}^s,\ c_2{}^s,\ \mathsf{sk} \cdot c_3{}^s,\ \mathsf{pk}_1{}^s,\ \mathsf{pk}_2{}^s, g^s)\ .$$

Commutative properties

Encrypt

To encrypt a message m:

$$c = (\mathsf{pk}_1{}^{r_1}, \mathsf{pk}_2{}^{r_2}, \mathcal{F}(m) \cdot g^{r_1+r_2})$$

Sign ∘ Encrypt

To sign a valid ciphertext c_1, c_2, c_3, one has simply to produce.

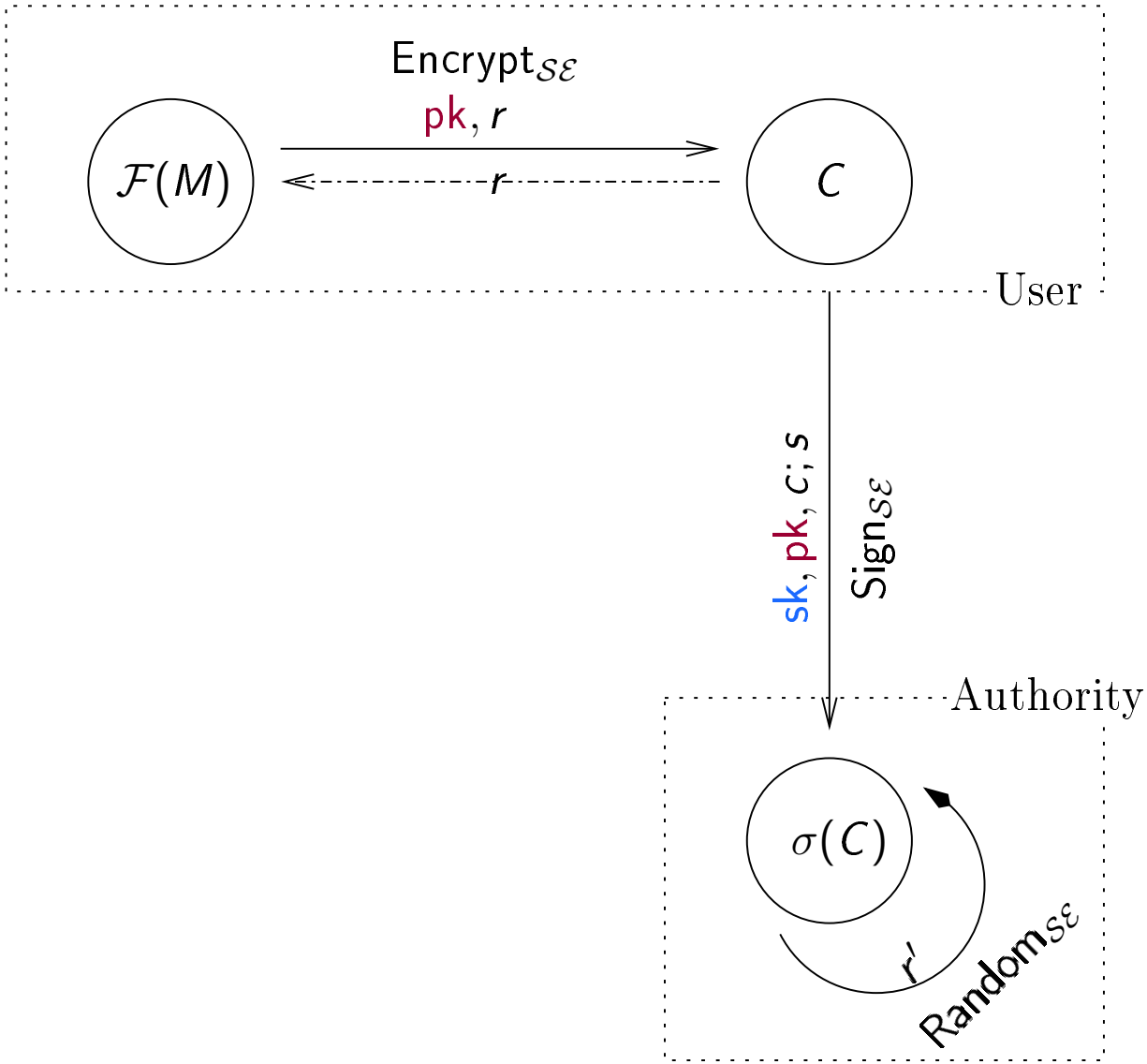
$$\sigma = (c_1{}^s,\ c_2{}^s,\ \mathsf{sk} \cdot c_3{}^s,\ \mathsf{pk}_1{}^s,\ \mathsf{pk}_2{}^s, g^s)\ .$$

Decrypt ∘ Sign ∘ Encrypt

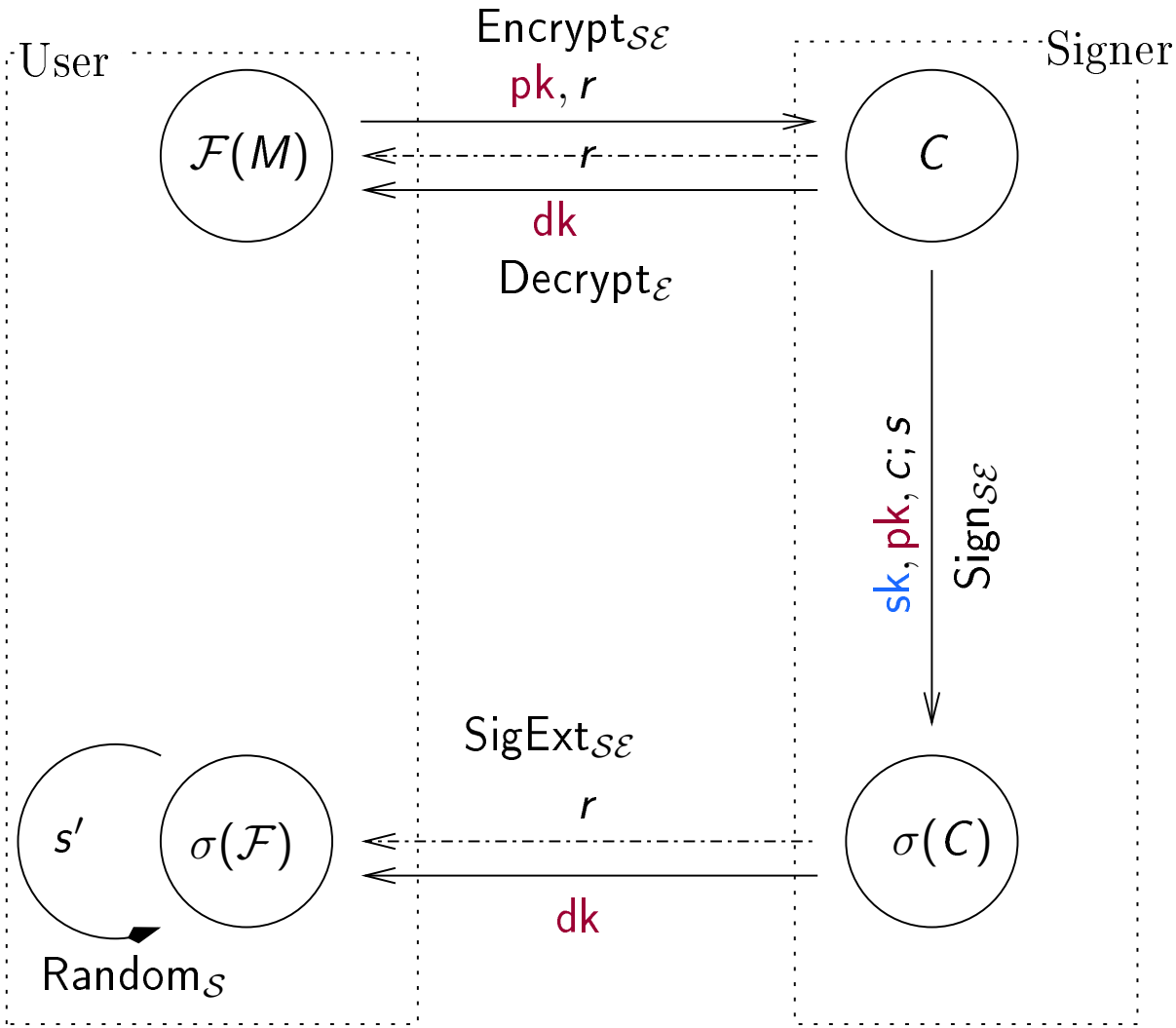
Using dk.

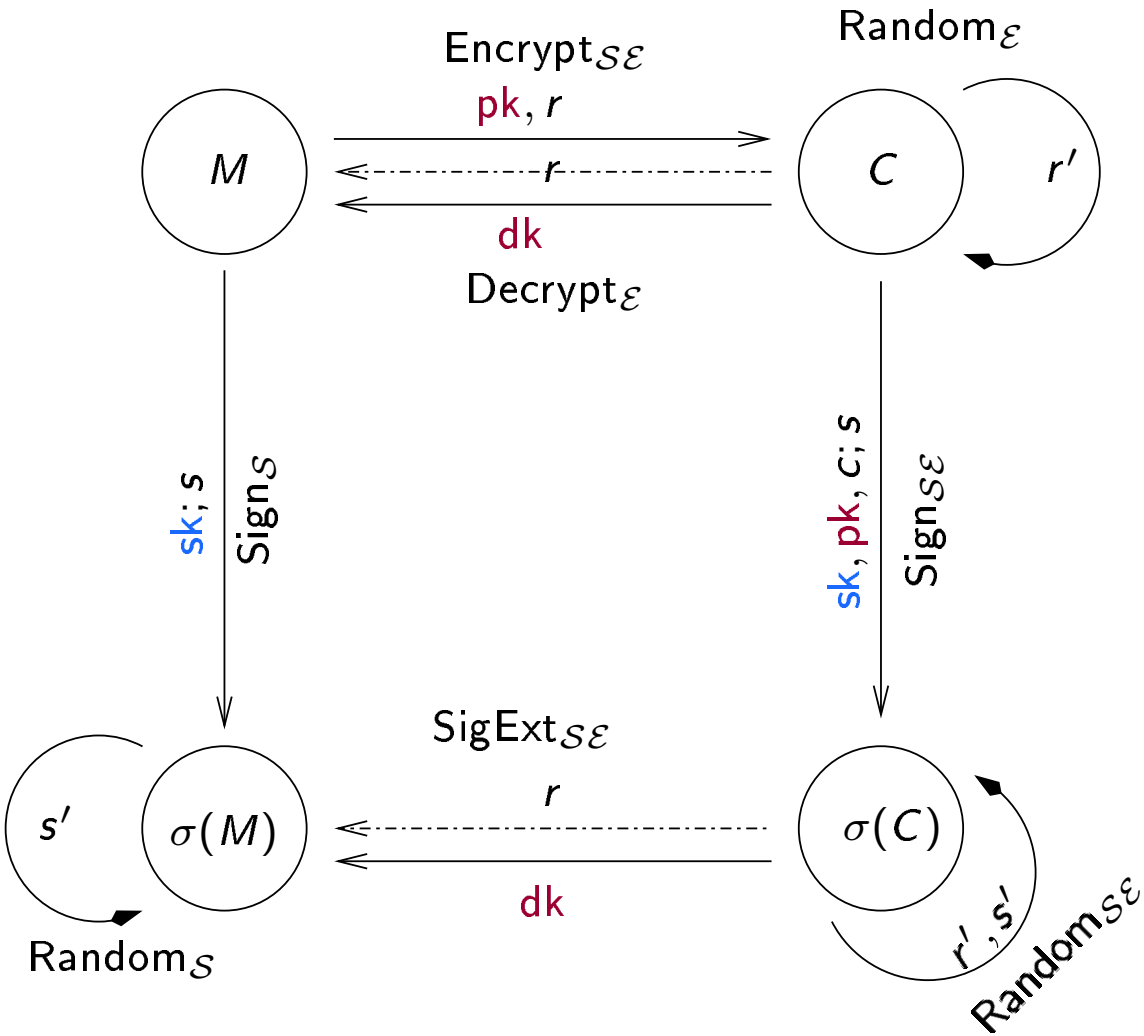
$$\sigma = (\sigma_3/\sigma_1^{\mathsf{dk}_1} \cdot \sigma_2^{\mathsf{dk}_2}, \sigma_6) = (\mathsf{sk} \cdot \mathcal{F}(m)^s, g^s)\ .$$

E-Voting


Encrypt$_{\mathcal{SE}}$
pk, r
$\mathcal{F}(M)$
r
C
User
sk, pk, c; s
Sign$_{\mathcal{SE}}$
Authority
$\sigma(C)$
r'
Random$_{\mathcal{SE}}$

Blind Signature




Encrypt$_{\mathcal{SE}}$
pk, r
r
dk
Decrypt$_{\mathcal{E}}$
Random$_{\mathcal{E}}$
M
C
r'
sk; s
Sign$_{\mathcal{S}}$
sk, pk, c; s
Sign$_{\mathcal{SE}}$
SigExt$_{\mathcal{SE}}$
r
dk
s'
$\sigma(M)$
$\sigma(C)$
Random$_{\mathcal{S}}$
r', s'
Random$_{\mathcal{SE}}$

- New primitive under standard assumption:

  Commutative property between Signature and Encryption

✓ One Round Blind Signature

✓ One Round Receipt Free E-Voting

Efficiency

- DLin + CDH : $9\ell + 33$ Group elements.
- SXDH + CDH$^+$: $6\ell + 15, 6\ell + 7$ Group elements.