

Threshold Signatures with Private Accountability

Dan Boneh¹ Chelsea Komlo²

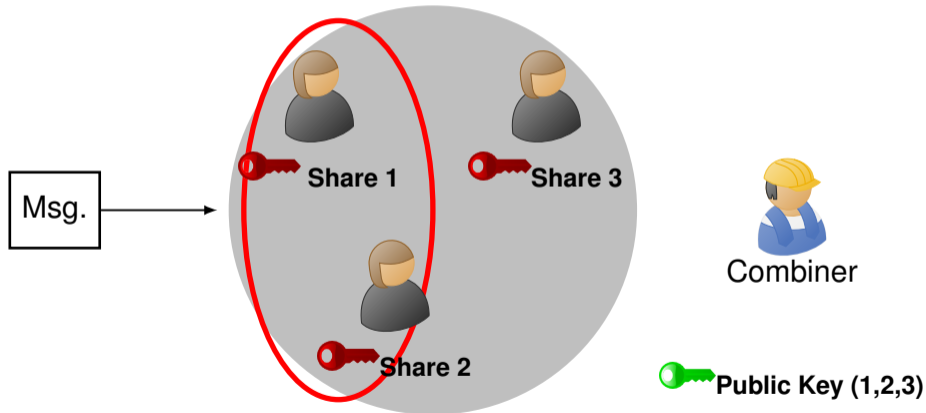
Stanford University

University of Waterloo

CRYPTO, August 13, 2022

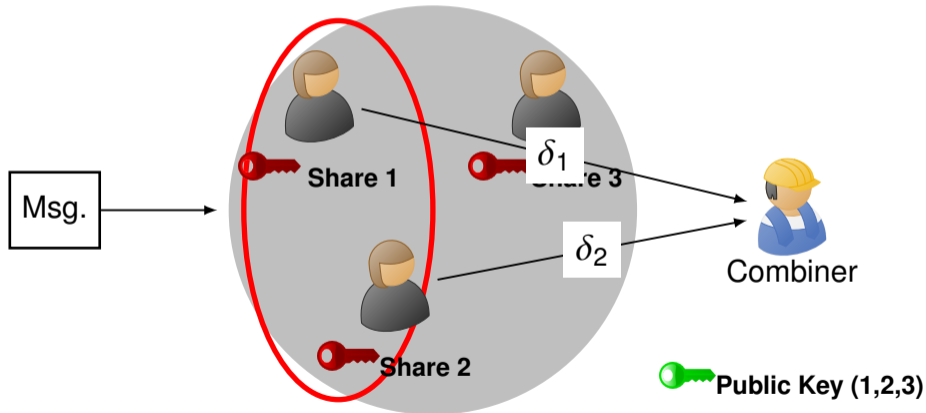
Threshold Signatures: Joint Public Key, Secret-Shared Private Key

($t = 2, n = 3$) Example



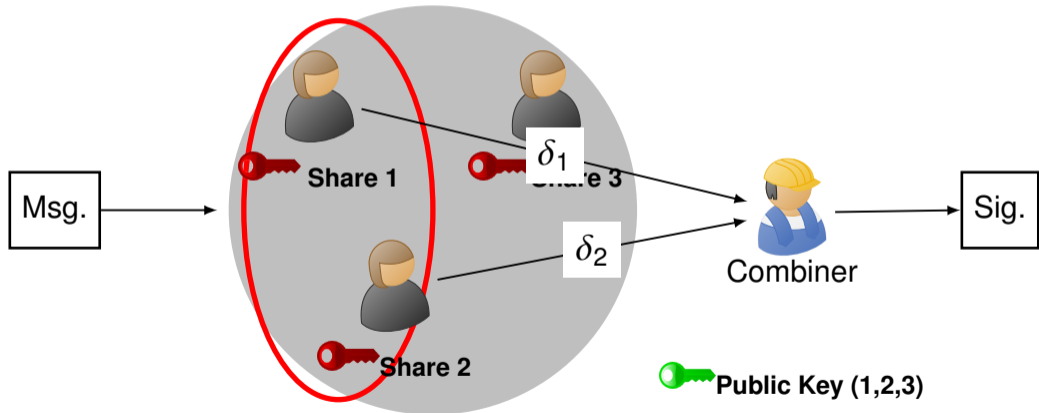
Threshold Signatures: Joint Public Key, Secret-Shared Private Key

($t = 2, n = 3$) Example



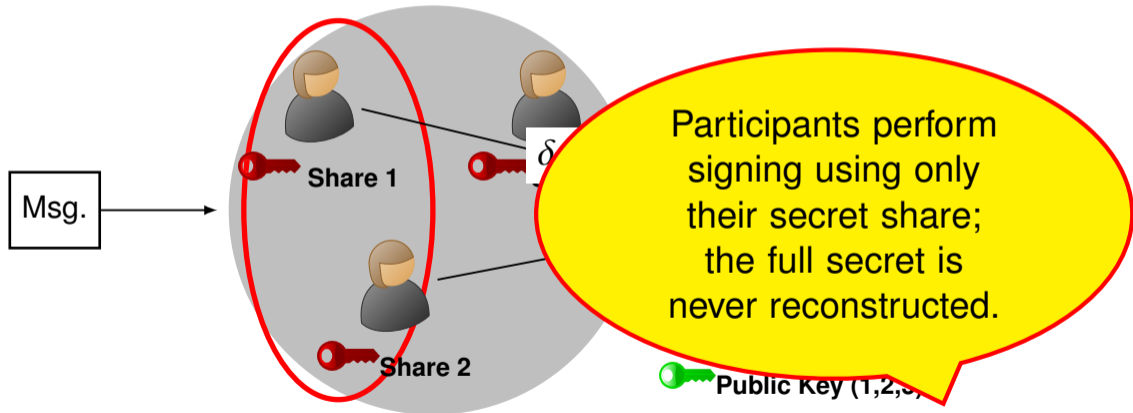
Threshold Signatures: Joint Public Key, Secret-Shared Private Key

($t = 2, n = 3$) Example



Threshold Signatures: Joint Public Key, Secret-Shared Private Key

($t = 2, n = 3$) Example



Existing Threshold Schemes

- ▶ In the literature, two types of (mutually exclusive) threshold schemes.
- ▶ **Full Privacy:** Private Threshold Scheme (PTS)
- ▶ **Full Accountability:** Accountable Threshold Scheme (ATS).
 - ▶ Also known as an Accountable Subgroup Multisignatures

Existing Threshold Schemes

- ▶ In the literature, two types of (mutually exclusive) threshold schemes.
- ▶ **Full Privacy:** Private Threshold Scheme (PTS)
- ▶ **Full Accountability:** Accountable Threshold Scheme (ATS).
 - ▶ Also known as an Accountable Subgroup Multisignatures

Existing Threshold Schemes

- ▶ In the literature, two types of (mutually exclusive) threshold schemes.
- ▶ **Full Privacy:** Private Threshold Scheme (PTS)
- ▶ **Full Accountability:** Accountable Threshold Scheme (ATS).
 - ▶ Also known as an Accountable Subgroup Multisignatures

Existing Threshold Schemes

- ▶ In the literature, two types of (mutually exclusive) threshold schemes.
- ▶ **Full Privacy:** Private Threshold Scheme (PTS)
- ▶ **Full Accountability:** Accountable Threshold Scheme (ATS).
 - ▶ Also known as an Accountable Subgroup Multisignatures

PTS versus ATS

	who learns signer quorum		who learns threshold	
	public	signers	public	signers
PTS	X	X	X	X
ATS	✓	✓	✓	✓

TAPS: Privacy and Accountability

- ▶ We introduce a *new* type of threshold signature.
- ▶ **Private *and* Accountable Threshold Signatures (TAPS)**
- ▶ Like a PTS, the public does not learn 1) the threshold, or 2) the signing quorum
- ▶ Like an ATS, the signing quorum can be recovered, but *only* by a designated entity.

TAPS: Privacy and Accountability

- ▶ We introduce a *new* type of threshold signature.
- ▶ **Private and Accountable Threshold Signatures (TAPS)**
- ▶ Like a PTS, the public does not learn 1) the threshold, or 2) the signing quorum
- ▶ Like an ATS, the signing quorum can be recovered, but *only* by a designated entity.

TAPS: Privacy and Accountability

- ▶ We introduce a *new* type of threshold signature.
- ▶ **Private *and* Accountable Threshold Signatures (TAPS)**
- ▶ Like a PTS, the public does not learn 1) the threshold, or 2) the signing quorum
- ▶ Like an ATS, the signing quorum can be recovered, but *only* by a designated entity.

TAPS: Privacy and Accountability

- ▶ We introduce a *new* type of threshold signature.
- ▶ **Private *and* Accountable Threshold Signatures (TAPS)**
- ▶ Like a PTS, the public does not learn 1) the threshold, or 2) the signing quorum
- ▶ Like an ATS, the signing quorum can be recovered, but *only* by a designated entity.

TAPS: Private and Accountable Threshold Signatures

Applications include:

- ▶ **Financial institutions:** prove or disprove issuance of funds.
- ▶ **Post-Compromise Accountability:** Allows for identification of malicious signers.

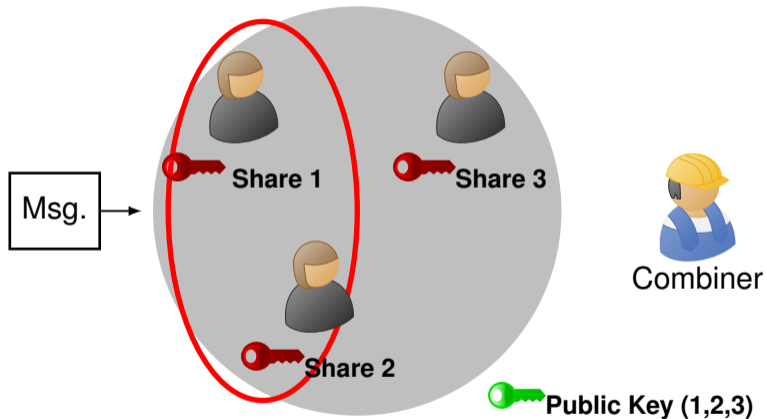
TAPS: Private and Accountable Threshold Signatures

Applications include:

- ▶ **Financial institutions:** prove or disprove issuance of funds.
- ▶ **Post-Compromise Accountability:** Allows for identification of malicious signers.

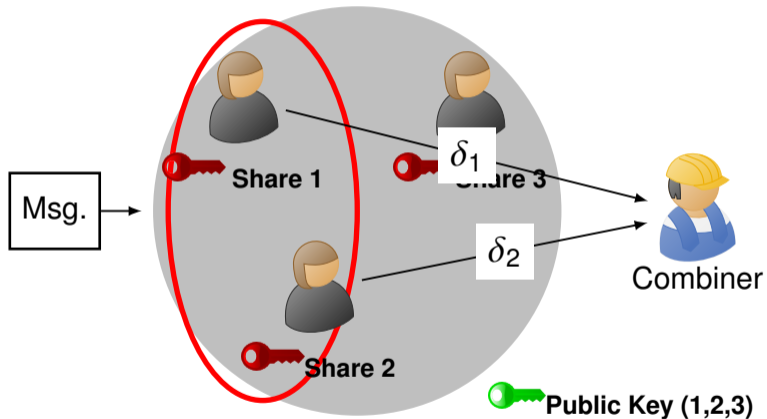
TAPS

($t = 2, n = 3$) Example



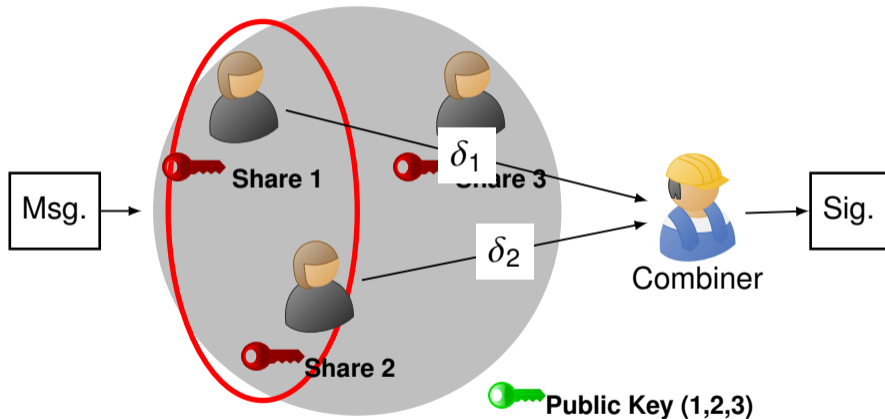
TAPS

($t = 2, n = 3$) Example



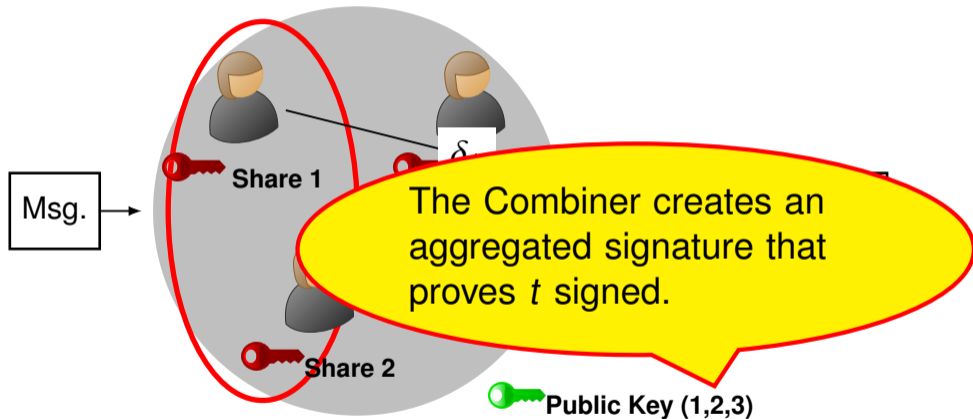
TAPS

($t = 2, n = 3$) Example



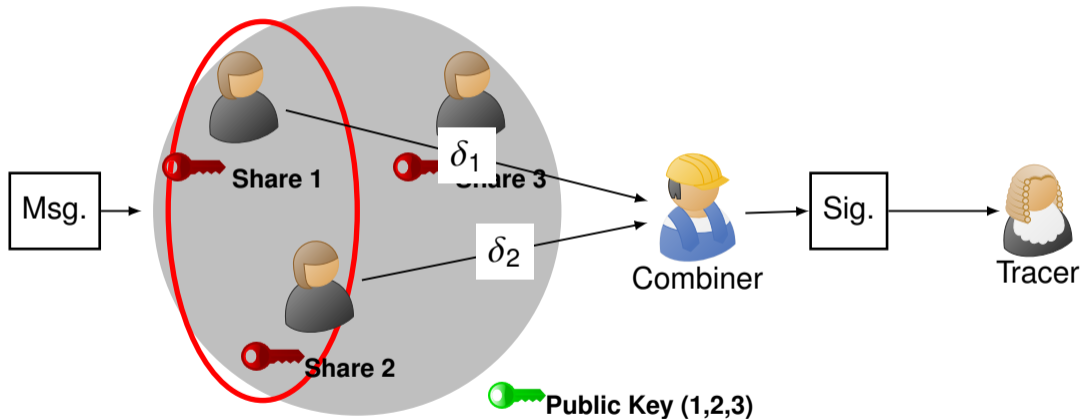
TAPS

($t = 2, n = 3$) Example



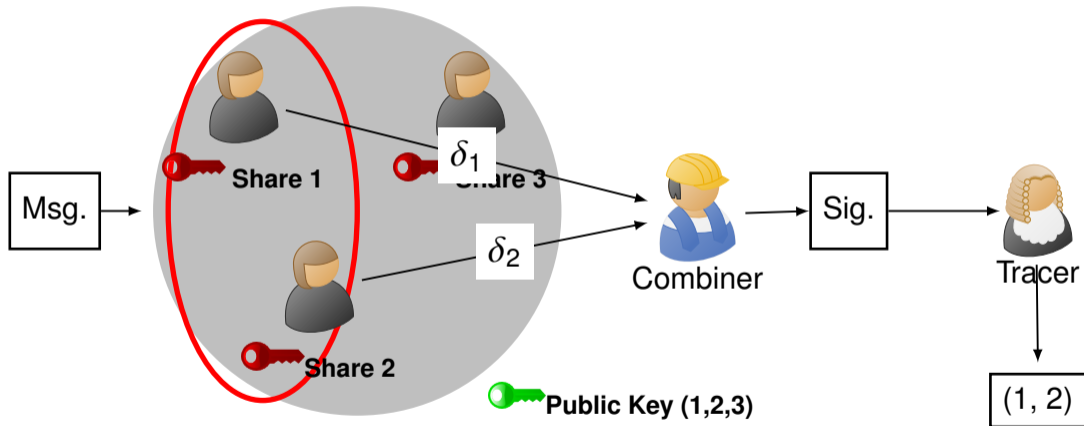
TAPS

($t = 2, n = 3$) Example



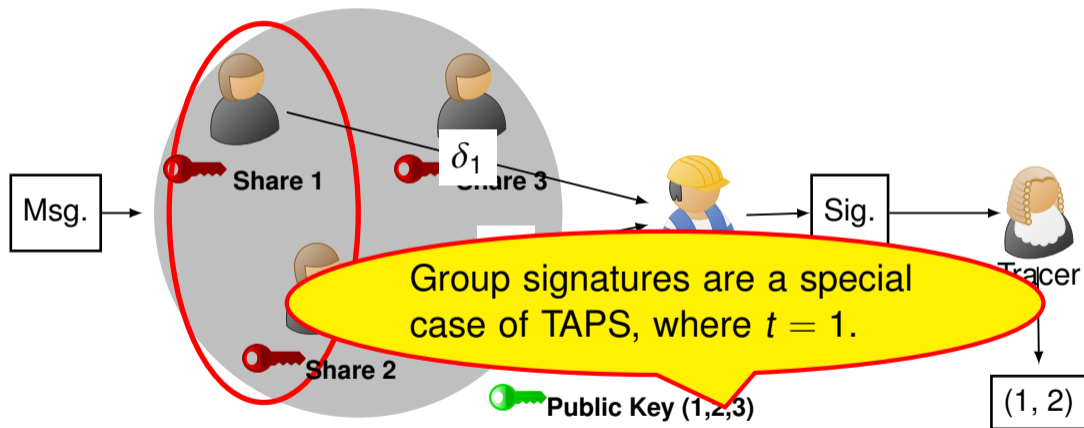
TAPS

($t = 2, n = 3$) Example



TAPS

($t = 2, n = 3$) Example



TAPS

Definition

A **private and accountable threshold signature** scheme, or **TAPS**, is a tuple of five polynomial time algorithms

$$\mathcal{S} = (\text{KeyGen}, \text{Sign}, \text{Combine}, \text{Verify}, \text{Trace})$$

A TAPS must be secure (unforgeable), private, and accountable.

TAPS

$$\text{KeyGen}(1^\lambda, n, t) \rightarrow (pk, (sk_1, \dots, sk_n), sk_c, sk_t)$$

- ▶ pk : The group's public key
 - ▶ (sk_1, \dots, sk_n) : Secret keys for each of the n participants.
 - ▶ sk_c : Secret key for the combiner
 - ▶ sk_t : Secret key for the tracer
-
- ▶ Could be performed in a distributed manner.

TAPS

$$\text{KeyGen}(1^\lambda, n, t) \rightarrow (pk, (sk_1, \dots, sk_n), sk_c, sk_t)$$

- ▶ pk : The group's public key
 - ▶ (sk_1, \dots, sk_n) : Secret keys for each of the n participants.
 - ▶ sk_c : Secret key for the combiner
 - ▶ sk_t : Secret key for the tracer
- ▶ Could be performed in a distributed manner.

TAPS

$$\text{KeyGen}(1^\lambda, n, t) \rightarrow (pk, (sk_1, \dots, sk_n), sk_c, sk_t)$$

- ▶ pk : The group's public key
 - ▶ (sk_1, \dots, sk_n) : Secret keys for each of the n participants.
 - ▶ sk_c : Secret key for the combiner
 - ▶ sk_t : Secret key for the tracer
- ▶ Could be performed in a distributed manner.

TAPS

$$\text{KeyGen}(1^\lambda, n, t) \rightarrow (pk, (sk_1, \dots, sk_n), sk_c, sk_t)$$

- ▶ pk : The group's public key
 - ▶ (sk_1, \dots, sk_n) : Secret keys for each of the n participants.
 - ▶ sk_c : Secret key for the combiner
 - ▶ sk_t : Secret key for the tracer
- ▶ Could be performed in a distributed manner.

TAPS

$$\text{KeyGen}(1^\lambda, n, t) \rightarrow (pk, (sk_1, \dots, sk_n), sk_c, sk_t)$$

- ▶ pk : The group's public key
 - ▶ (sk_1, \dots, sk_n) : Secret keys for each of the n participants.
 - ▶ sk_c : Secret key for the combiner
 - ▶ sk_t : Secret key for the tracer
-
- ▶ Could be performed in a distributed manner.

TAPS

$Sign(sk_i, m, C) \rightarrow \delta_i$

- ▶ m : Message to be signed
- ▶ C : Coalition of signers
- ▶ δ_i : Partial signature for participant i

$Combine(sk_C, m, C, \{\delta_i\}_{i \in C}) \rightarrow \sigma$

- ▶ Outputs σ , a TAPS signature

TAPS

$Sign(sk_i, m, C) \rightarrow \delta_i$

- ▶ m : Message to be signed
- ▶ C : Coalition of signers
- ▶ δ_i : Partial signature for participant i

$Combine(sk_C, m, C, \{\delta_i\}_{i \in C}) \rightarrow \sigma$

- ▶ Outputs σ , a TAPS signature

TAPS

$Sign(sk_i, m, C) \rightarrow \delta_i$

- ▶ m : Message to be signed
- ▶ C : Coalition of signers
- ▶ δ_i : Partial signature for participant i

$Combine(sk_C, m, C, \{\delta_i\}_{i \in C}) \rightarrow \sigma$

- ▶ Outputs σ , a TAPS signature

TAPS

$Sign(sk_i, m, C) \rightarrow \delta_i$

- ▶ m : Message to be signed
- ▶ C : Coalition of signers
- ▶ δ_i : Partial signature for participant i

$Combine(sk_c, m, C, \{\delta_i\}_{i \in C}) \rightarrow \sigma$

- ▶ Outputs σ , a TAPS signature

TAPS

$Sign(sk_i, m, C) \rightarrow \delta_i$

- ▶ m : Message to be signed
- ▶ C : Coalition of signers
- ▶ δ_i : Partial signature for participant i

$Combine(sk_C, m, C, \{\delta_i\}_{i \in C}) \rightarrow \sigma$

- ▶ Outputs σ , a TAPS signature

TAPS

$Verify(pk, m, \sigma) \rightarrow 0/1:$

- ▶ Outputs a bit indicating if σ is valid for pk, m

$Trace(sk_t, m, \sigma) \rightarrow C/fail:$

- ▶ Outputs either the coalition of signers or fails.

TAPS

$Verify(pk, m, \sigma) \rightarrow 0/1:$

- ▶ Outputs a bit indicating if σ is valid for pk, m

$Trace(sk_t, m, \sigma) \rightarrow C/fail:$

- ▶ Outputs either the coalition of signers or fails.

TAPS

$Verify(pk, m, \sigma) \rightarrow 0/1:$

- ▶ Outputs a bit indicating if σ is valid for pk, m

$Trace(sk_t, m, \sigma) \rightarrow C/fail:$

- ▶ Outputs either the coalition of signers or fails.

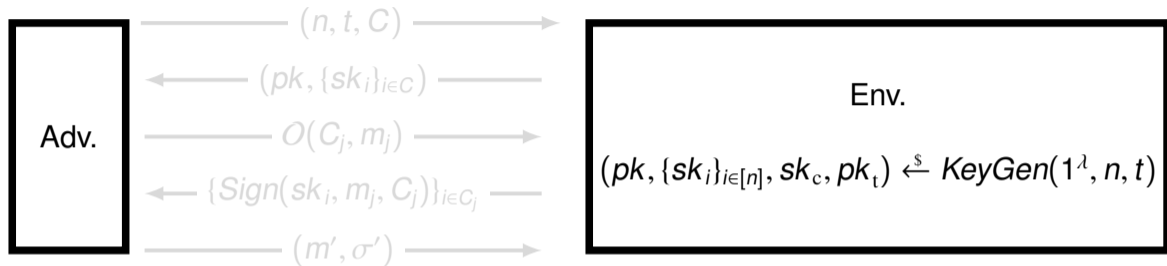
Unforgeability and Accountability

- ▶ **Unforgeability.** Adversary cannot output a valid signature when controlling fewer than t parties.
- ▶ **Accountability.** Adversary cannot output a valid signature that traces to an honest non-signer.

Unforgeability and Accountability

- ▶ **Unforgeability.** Adversary cannot output a valid signature when controlling fewer than t parties.
- ▶ **Accountability.** Adversary cannot output a valid signature that traces to an honest non-signer.

Unforgeability and Accountability

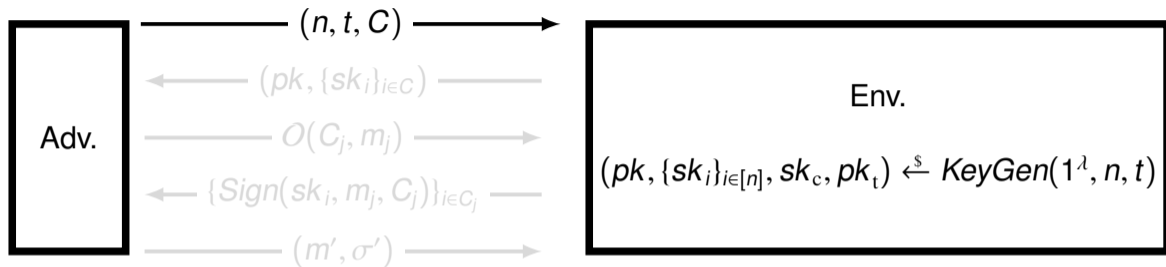


Adv wins if:

- (1) It produces a valid signature and $|C| < t$ (**unforgeability**), or
- (2) It outputs a valid signature that traces to an honest non-signer $i \notin C$ (**accountability**).

TAPS is *unforgeable* and *accountable* if $\Pr[\text{Adv wins}]$ is negligible.

Unforgeability and Accountability

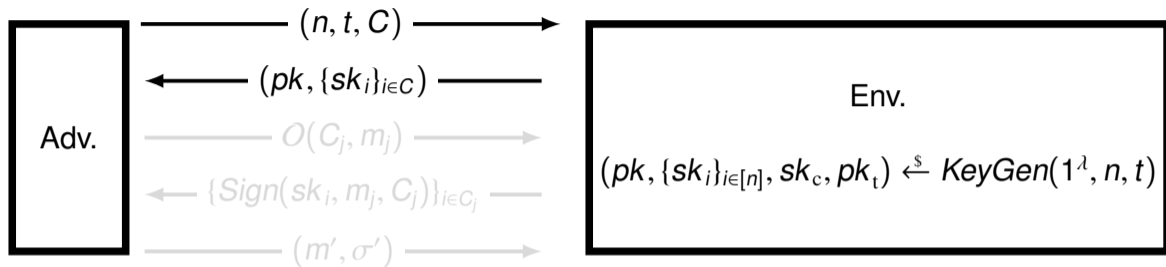


Adv wins if:

- (1) It produces a valid signature and $|C| < t$ (**unforgeability**), or
- (2) It outputs a valid signature that traces to an honest non-signer $i \notin C$ (**accountability**).

TAPS is *unforgeable* and *accountable* if $\Pr[\text{Adv wins}]$ is negligible.

Unforgeability and Accountability

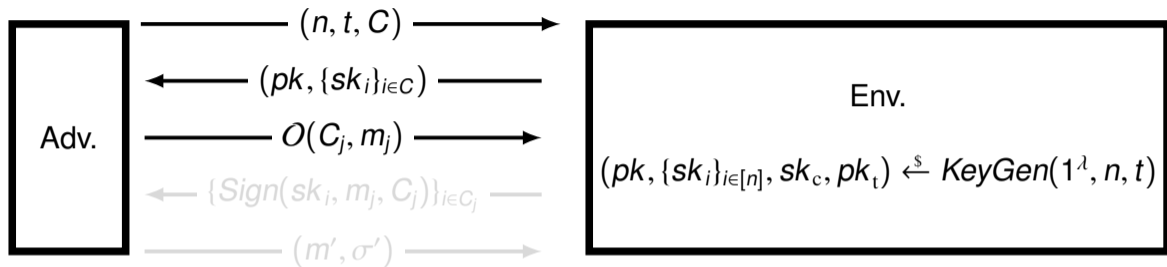


Adv wins if:

- (1) It produces a valid signature and $|C| < t$ (**unforgeability**), or
- (2) It outputs a valid signature that traces to an honest non-signer $i \notin C$ (**accountability**).

TAPS is *unforgable* and *accountable* if $\Pr[\text{Adv wins}]$ is negligible.

Unforgeability and Accountability

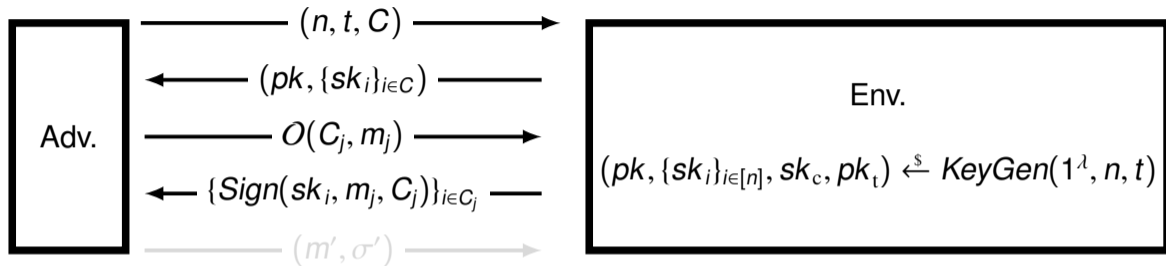


Adv wins if:

- (1) It produces a valid signature and $|C| < t$ (**unforgeability**), or
- (2) It outputs a valid signature that traces to an honest non-signer $i \notin C$ (**accountability**).

TAPS is *unforgeable* and *accountable* if $\Pr[\text{Adv wins}]$ is negligible.

Unforgeability and Accountability

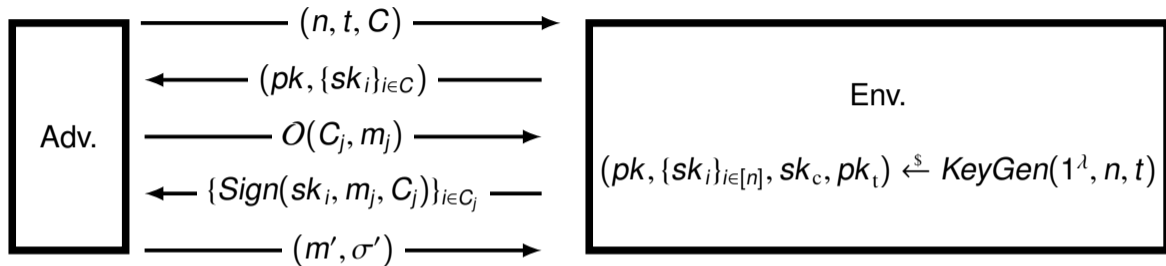


Adv wins if:

- (1) It produces a valid signature and $|C| < t$ (**unforgeability**), or
- (2) It outputs a valid signature that traces to an honest non-signer $i \notin C$ (**accountability**).

TAPS is *unforgeable* and *accountable* if $\Pr[\text{Adv wins}]$ is negligible.

Unforgeability and Accountability

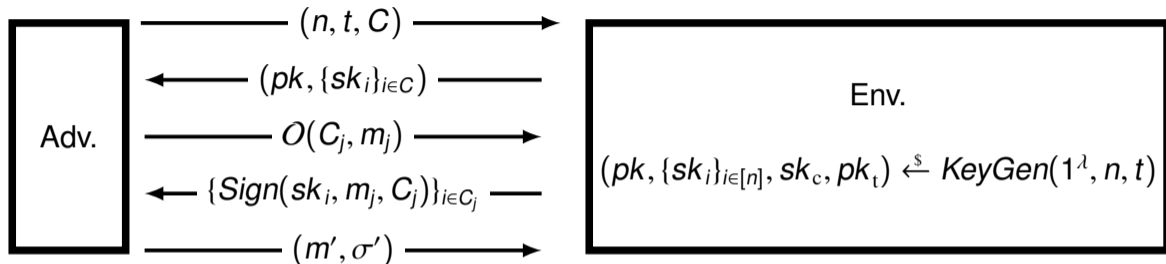


Adv wins if:

- (1) It produces a valid signature and $|C| < t$ (**unforgeability**), or
- (2) It outputs a valid signature that traces to an honest non-signer $i \notin C$ (**accountability**).

TAPS is *unforgeable* and *accountable* if $\Pr[\text{Adv wins}]$ is negligible.

Unforgeability and Accountability

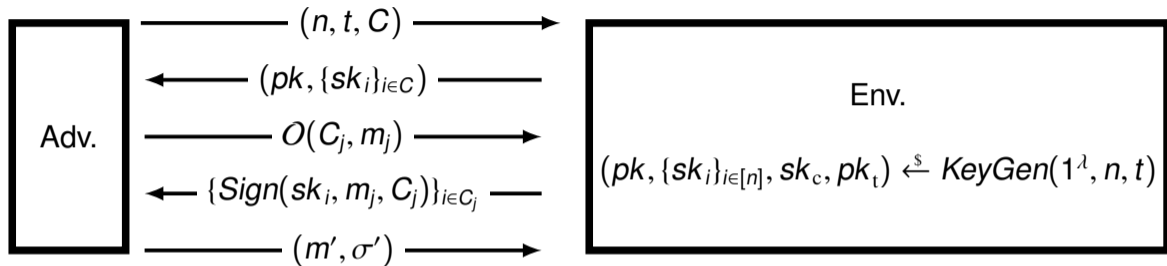


Adv wins if:

- (1) It produces a valid signature and $|C| < t$ (**unforgeability**), or
- (2) It outputs a valid signature that traces to an honest non-signer $i \notin C$ (**accountability**).

TAPS is *unforgeable* and *accountable* if $\Pr[\text{Adv wins}]$ is negligible.

Unforgeability and Accountability

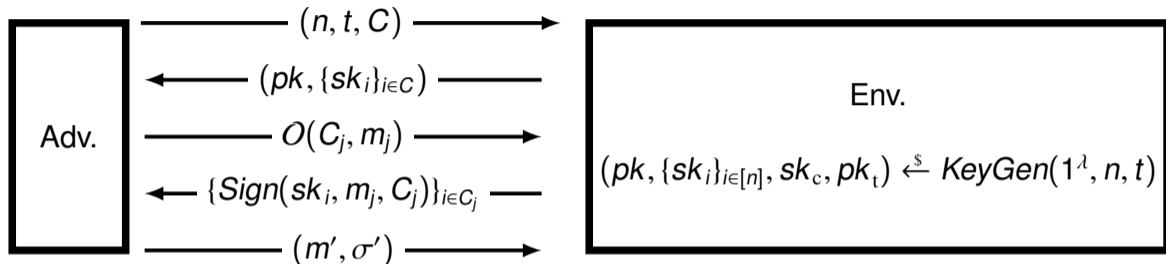


Adv wins if:

- (1) It produces a valid signature and $|C| < t$ (**unforgeability**), or
- (2) It outputs a valid signature that traces to an honest non-signer $i \notin C$ (**accountability**).

TAPS is *unforgeable* and *accountable* if $\Pr[\text{Adv wins}]$ is negligible.

Unforgeability and Accountability



Adv wins if:

- (1) It produces a valid signature and $|C| < t$ (**unforgeability**), or
- (2) It outputs a valid signature that traces to an honest non-signer $i \notin C$ (**accountability**).

TAPS is *unforgable* and *accountable* if $\Pr[\text{Adv wins}]$ is negligible.

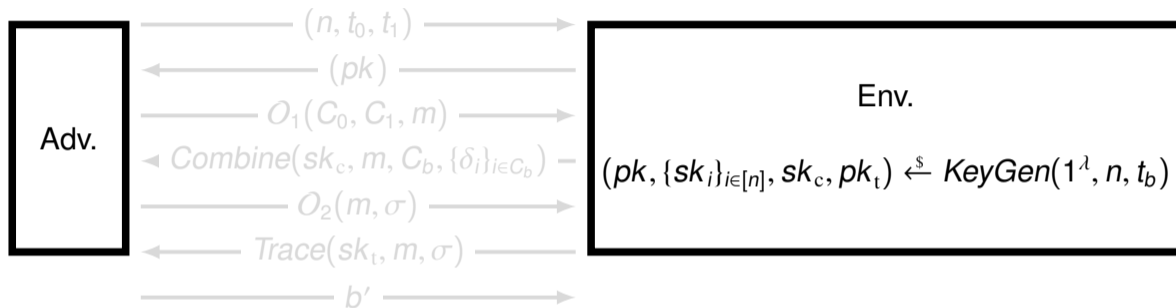
Privacy

- ▶ **Privacy against the public.** The TAPS signature reveals nothing about the threshold or quorum of signers to the *public*.
- ▶ **Privacy against signers.** The TAPS signature reveals nothing about the quorum to *signers*.

Privacy

- ▶ **Privacy against the public.** The TAPS signature reveals nothing about the threshold or quorum of signers to the *public*.
- ▶ **Privacy against signers.** The TAPS signature reveals nothing about the quorum to *signers*.

Privacy Against The Public

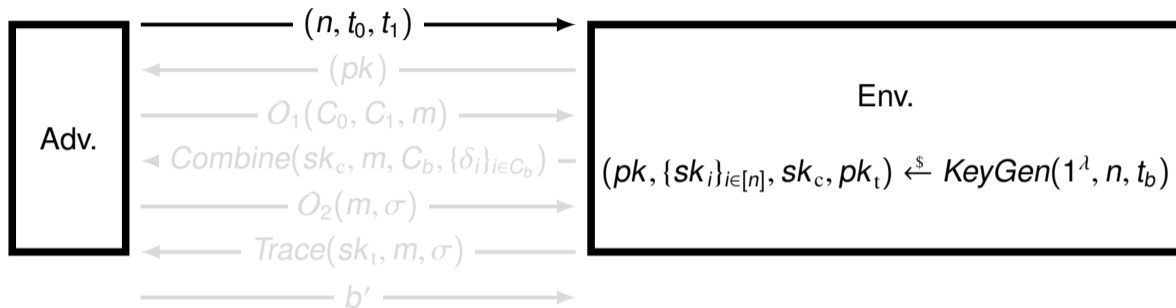


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against the public. Adversary wins if it can gain any information about t .

TAPS is *private against the public* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against The Public

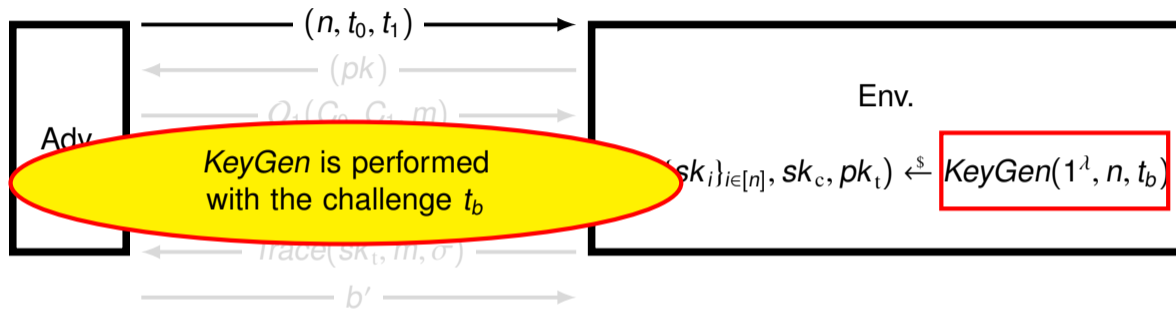


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against the public. Adversary wins if it can gain any information about t .

TAPS is *private against the public* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against The Public

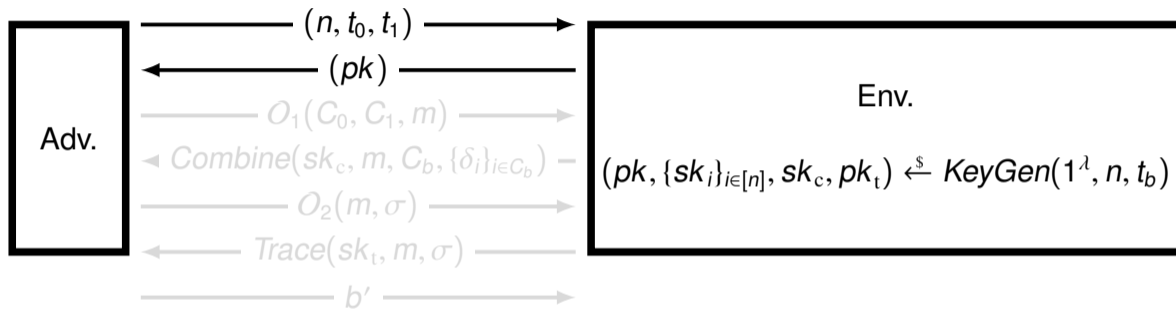


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against the public. Adversary wins if it can gain any information about t .

TAPS is *private against the public* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against The Public

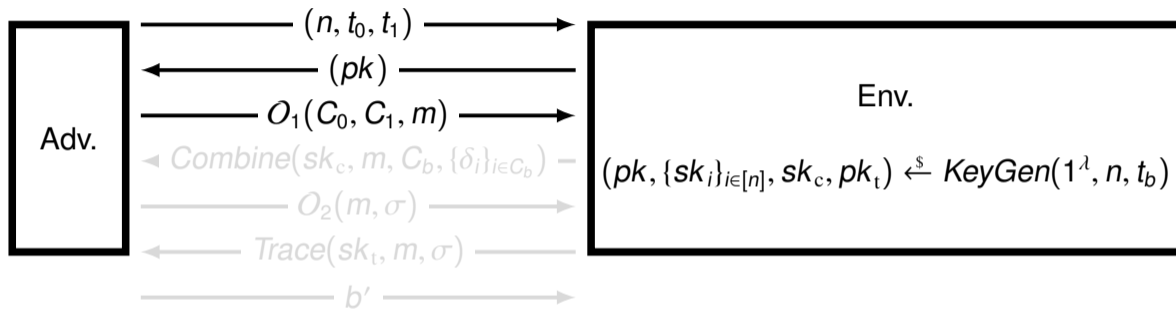


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against the public. Adversary wins if it can gain any information about t .

TAPS is *private against the public* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against The Public

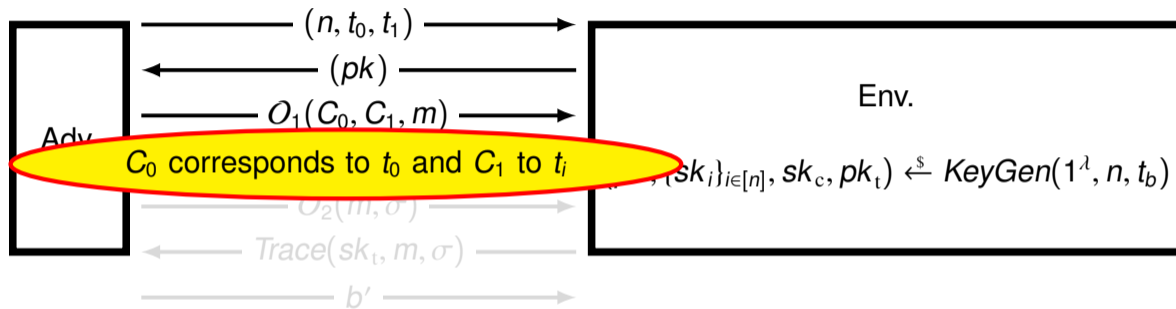


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against the public. Adversary wins if it can gain any information about t .

TAPS is *private against the public* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against The Public

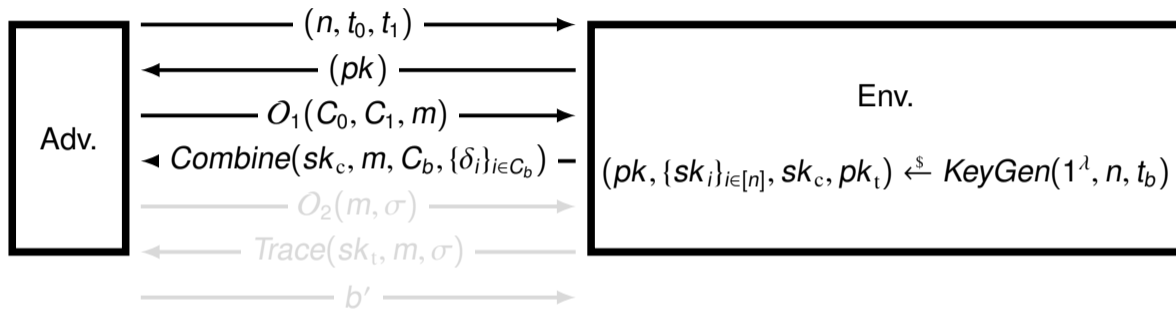


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against the public. Adversary wins if it can gain any information about t .

TAPS is *private against the public* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against The Public

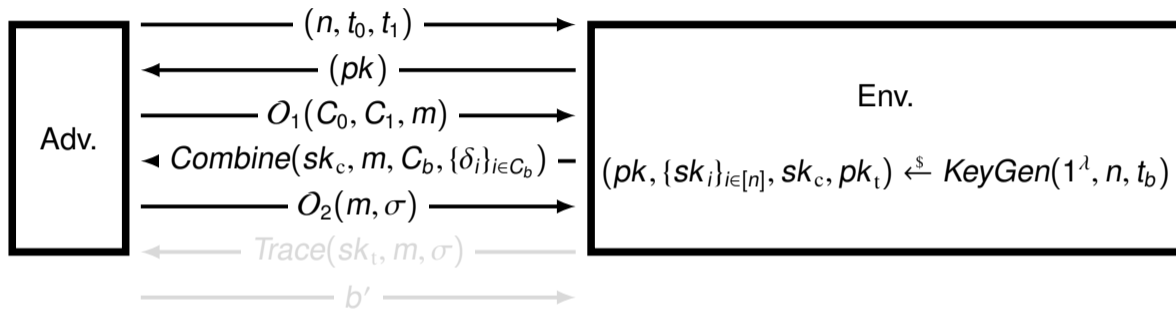


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against the public. Adversary wins if it can gain any information about t .

TAPS is *private against the public* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against The Public

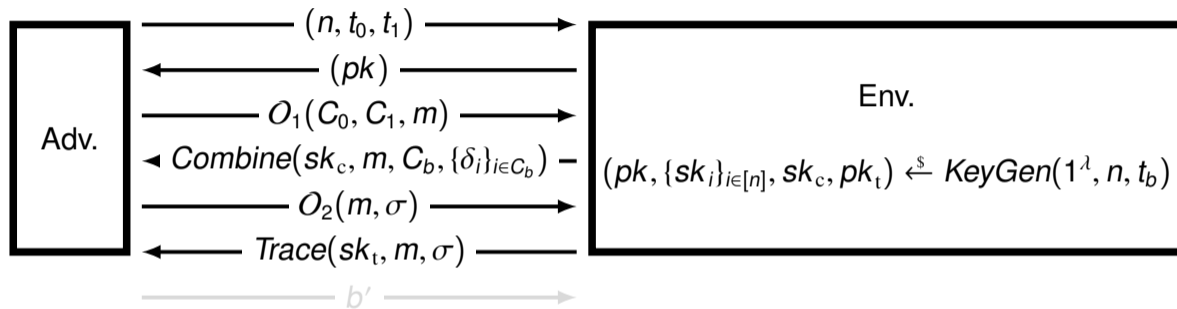


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against the public. Adversary wins if it can gain any information about t .

TAPS is *private against the public* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against The Public

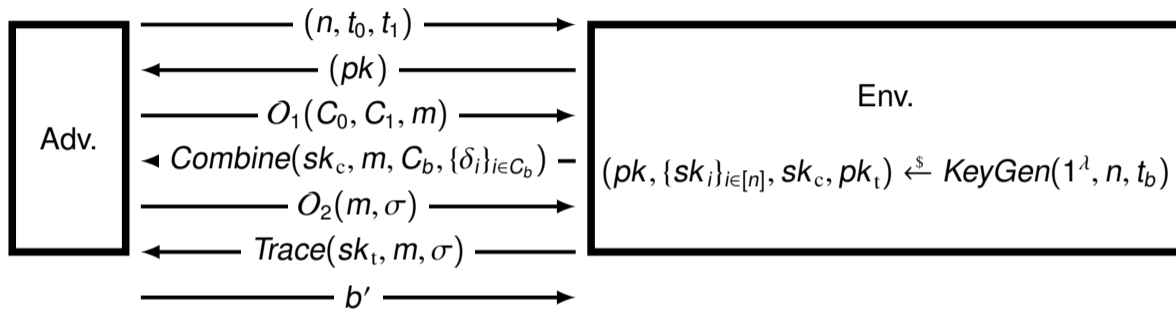


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against the public. Adversary wins if it can gain any information about t .

TAPS is *private against the public* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against The Public

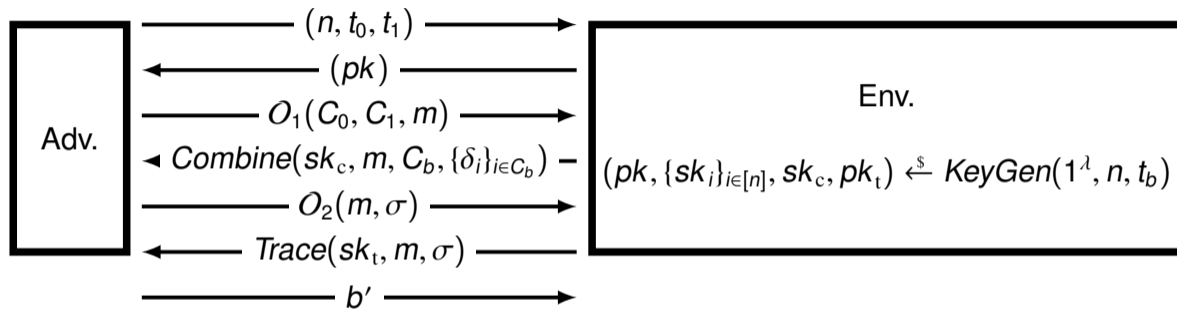


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against the public. Adversary wins if it can gain any information about t .

TAPS is *private against the public* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against The Public

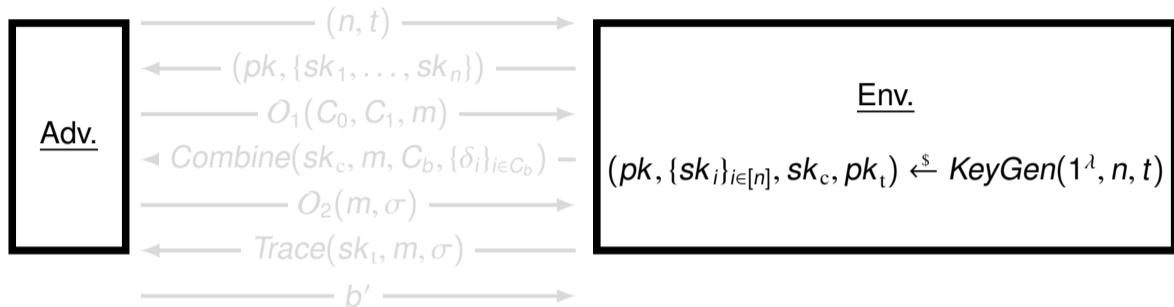


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against the public. Adversary wins if it can gain any information about t .

TAPS is *private against the public* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against Signers

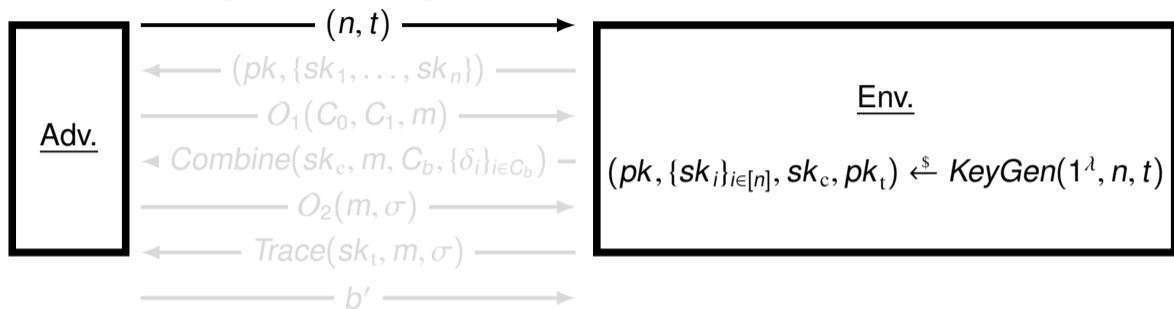


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against signers. Adv. wins if it can gain information about the quorum of signers.

TAPS is *private against signers* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against Signers

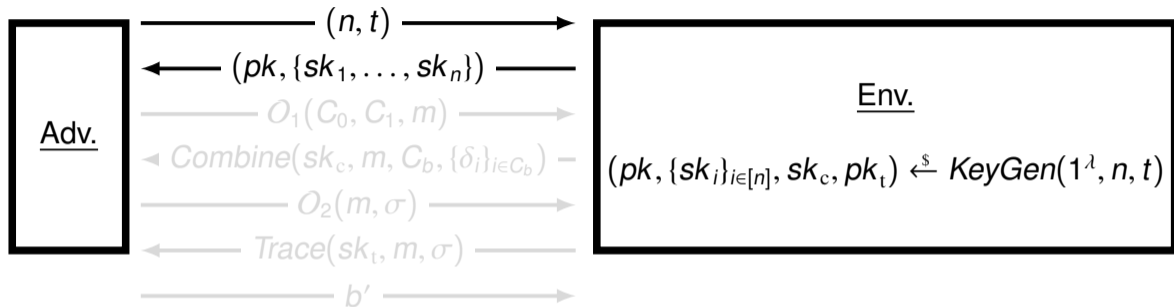


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against signers. Adv. wins if it can gain information about the quorum of signers.

TAPS is *private against signers* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against Signers

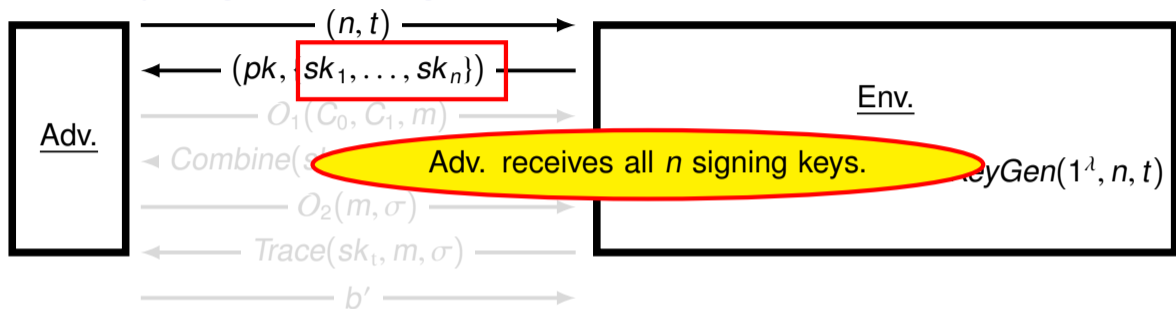


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against signers. Adv. wins if it can gain information about the quorum of signers.

TAPS is *private against signers* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against Signers

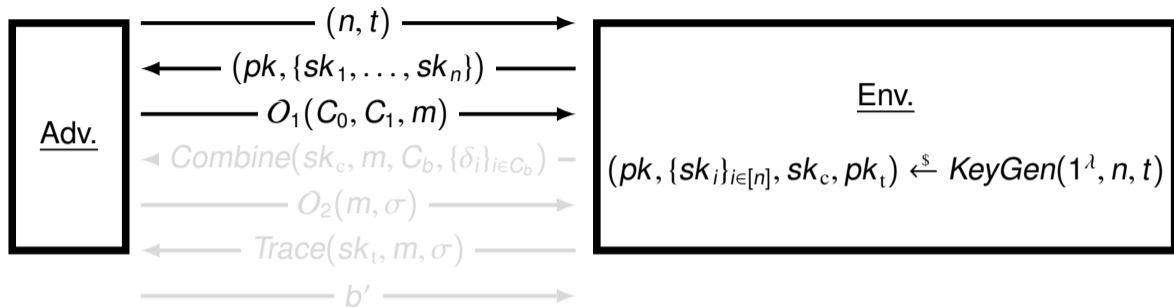


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against signers. Adv. wins if it can gain information about the quorum of signers.

TAPS is *private against signers* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against Signers

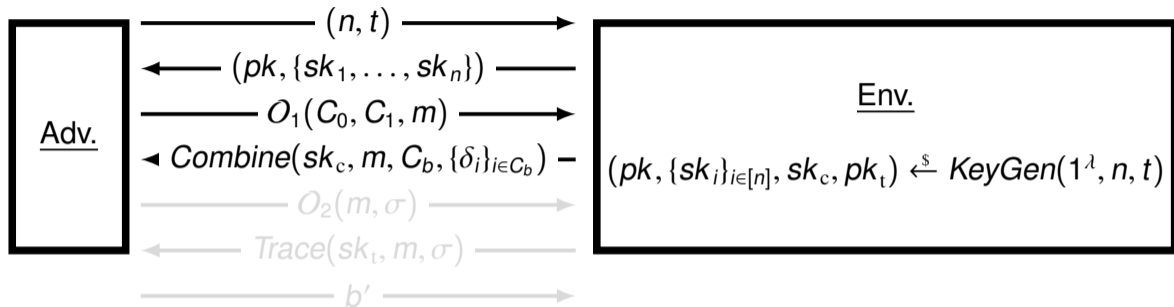


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against signers. Adv. wins if it can gain information about the quorum of signers.

TAPS is *private against signers* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against Signers

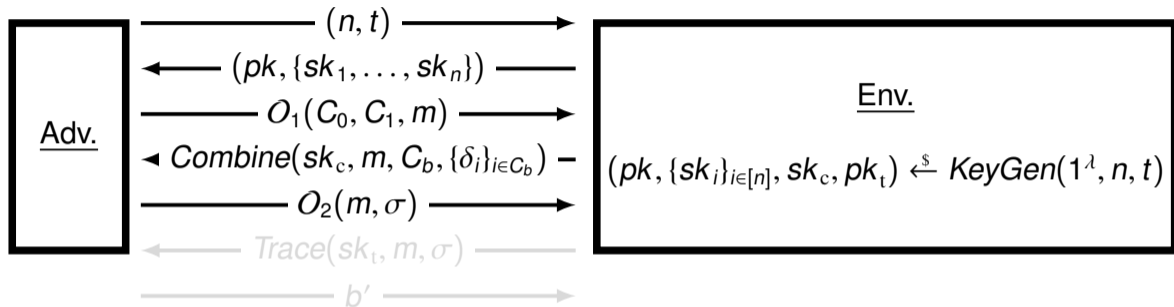


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against signers. Adv. wins if it can gain information about the quorum of signers.

TAPS is *private against signers* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against Signers

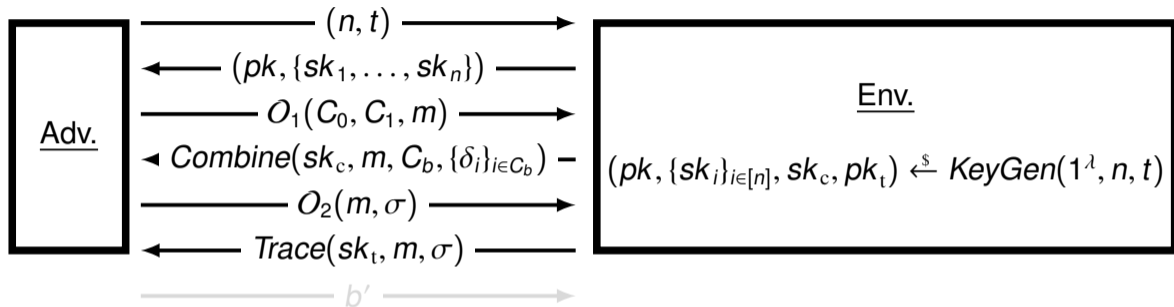


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against signers. Adv. wins if it can gain information about the quorum of signers.

TAPS is *private against signers* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against Signers

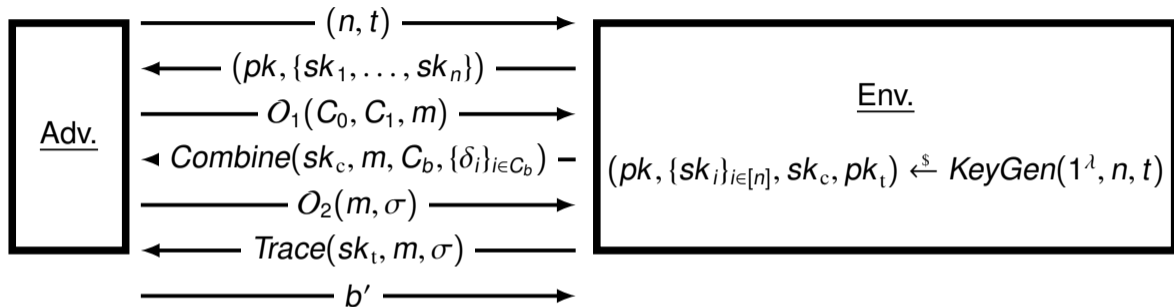


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against signers. Adv. wins if it can gain information about the quorum of signers.

TAPS is *private against signers* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against Signers

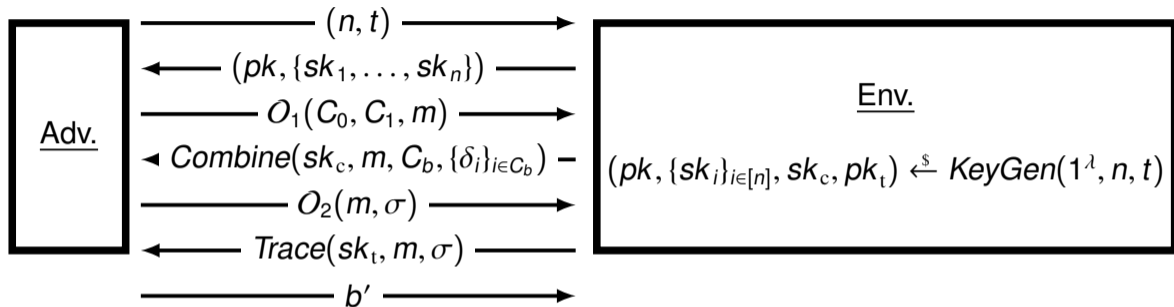


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against signers. Adv. wins if it can gain information about the quorum of signers.

TAPS is *private against signers* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against Signers

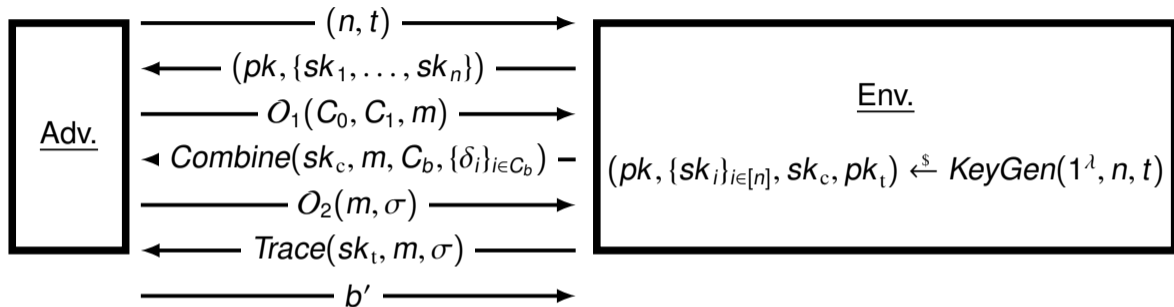


Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against signers. Adv. wins if it can gain information about the quorum of signers.

TAPS is *private against signers* if $\Pr[\text{Adv wins}]$ is negligible.

Privacy Against Signers



Restriction: Inputs to O_2 cannot be outputs from O_1 .

Privacy against signers. Adv. wins if it can gain information about the quorum of signers.

TAPS is *private against signers* if $\Pr[\text{Adv wins}]$ is negligible.

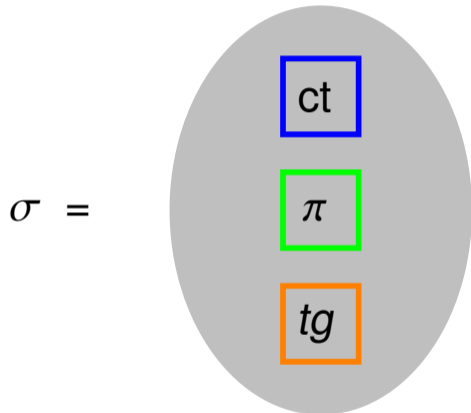
Generic TAPS

- ▶ TAPS signature σ proves in generic zero knowledge that an encrypted ATS signature is valid.
- ▶ TAPS public key contains a *commitment* to the threshold.

Generic TAPS

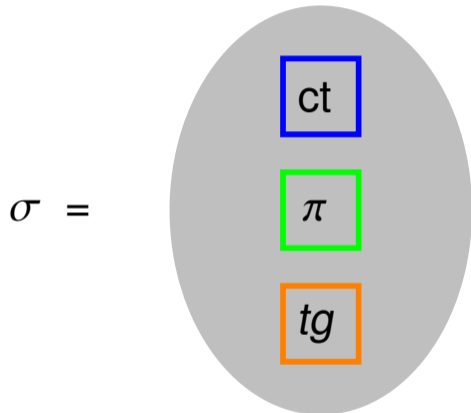
- ▶ TAPS signature σ proves in generic zero knowledge that an encrypted ATS signature is valid.
- ▶ TAPS public key contains a *commitment* to the threshold.

Generic TAPS: Signature



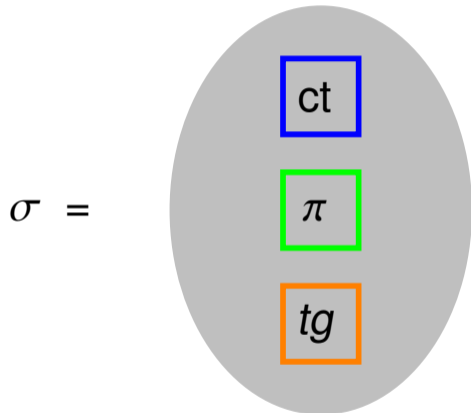
- ▶ ct : Public-key encryption of an ATS signature to the tracer.
- ▶ π : zero-knowledge proof that the ATS is a valid signature on m
- ▶ tg : The combiner's signature on (m, ct, π)

Generic TAPS: Signature



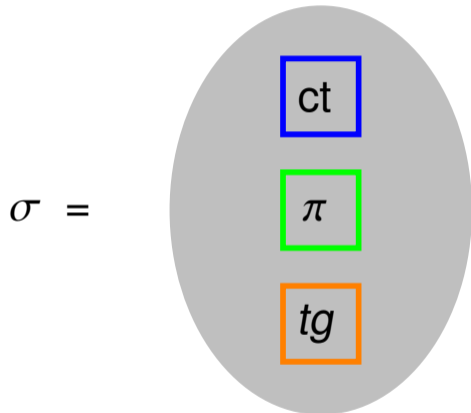
- ▶ ct : Public-key encryption of an ATS signature to the tracer.
- ▶ π : zero-knowledge proof that the ATS is a valid signature on m
- ▶ tg : The combiner's signature on (m, ct, π)

Generic TAPS: Signature



- ▶ ct : Public-key encryption of an ATS signature to the tracer.
- ▶ π : zero-knowledge proof that the ATS is a valid signature on m
- ▶ tg : The combiner's signature on (m, ct, π)

Generic TAPS: Signature



- ▶ ct : Public-key encryption of an ATS signature to the tracer.
- ▶ π : zero-knowledge proof that the ATS is a valid signature on m
- ▶ tg : The combiner's signature on (m, ct, π)

On Schnorr and Generic Zero Knowledge

- ▶ The generic scheme requires proving the signature is valid in generic zero knowledge.
- ▶ Schnorr requires proving the output from a hash function is derived correctly, which is expensive in generic zero knowledge.
- ▶ **Goal:** Find a simpler, more efficient solution.

On Schnorr and Generic Zero Knowledge

- ▶ The generic scheme requires proving the signature is valid in generic zero knowledge.
- ▶ Schnorr requires proving the output from a hash function is derived correctly, which is expensive in generic zero knowledge.
- ▶ **Goal:** Find a simpler, more efficient solution.

On Schnorr and Generic Zero Knowledge

- ▶ The generic scheme requires proving the signature is valid in generic zero knowledge.
- ▶ Schnorr requires proving the output from a hash function is derived correctly, which is expensive in generic zero knowledge.
- ▶ **Goal:** Find a simpler, more efficient solution.

Schnorr TAPS: Solution

- ▶ Recall a Schnorr signature σ :

$$\sigma = (R, \text{ // This is a commitment to a nonce } z) \text{ // This is the response}$$

- ▶ The verifier derives $c = H(R, m)$, and checks $R \stackrel{?}{=} g^z \cdot pk^{-c}$.
- ▶ **Insight:** Publishing R in the clear does not hurt privacy! Only z must be protected.

Schnorr TAPS: Solution

- ▶ Recall a Schnorr signature σ :

$$\sigma = (R, \text{ // This is a commitment to a nonce } z) \text{ // This is the response}$$

- ▶ The verifier derives $c = H(R, m)$, and checks $R \stackrel{?}{=} g^z \cdot pk^{-c}$.
- ▶ **Insight:** Publishing R in the clear does not hurt privacy! Only z must be protected.

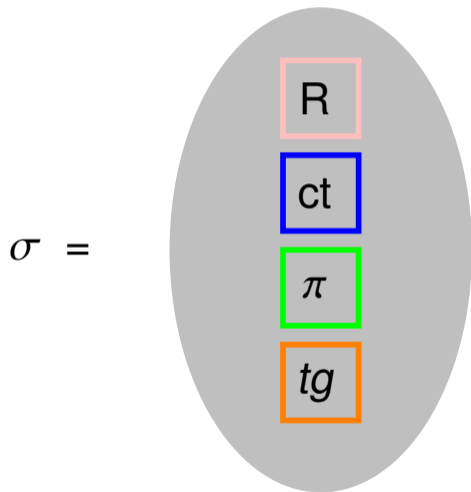
Schnorr TAPS: Solution

- ▶ Recall a Schnorr signature σ :

$$\sigma = (R, \text{ // This is a commitment to a nonce } z) \text{ // This is the response}$$

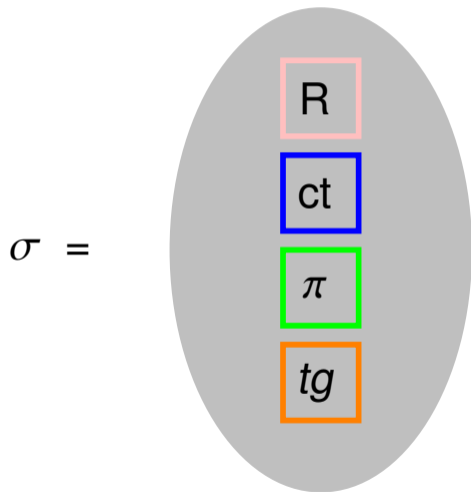
- ▶ The verifier derives $c = H(R, m)$, and checks $R \stackrel{?}{=} g^z \cdot pk^{-c}$.
- ▶ **Insight:** Publishing R in the clear does not hurt privacy! Only z must be protected.

Schnorr TAPS



- ▶ Improved efficiency and simplicity over generic zero knowledge.
- ▶ Verifier derives the challenge directly.

Schnorr TAPS



- ▶ Improved efficiency and simplicity over generic zero knowledge.
- ▶ Verifier derives the challenge directly.

Schnorr TAPS Relation

$$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$$

- ▶ g, h are generators whose dlog relation is unknown
- ▶ (T_0, T_1) is an ElGamal commitment to t
- ▶ R is the Schnorr commitment
- ▶ c is the Schnorr challenge
- ▶ $ct = (c_0, c_1)$ is the encrypted ATS signature z
- ▶ ρ, ψ are randomizers
- ▶ b_1, \dots, b_n are bits indicating which signers participated

Schnorr TAPS Relation

$$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$$

- ▶ g, h are generators whose dlog relation is unknown
- ▶ (T_0, T_1) is an ElGamal commitment to t
- ▶ R is the Schnorr commitment
- ▶ c is the Schnorr challenge
- ▶ $ct = (c_0, c_1)$ is the encrypted ATS signature z
- ▶ ρ, ψ are randomizers
- ▶ b_1, \dots, b_n are bits indicating which signers participated

Schnorr TAPS Relation

$$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$$

- ▶ g, h are generators whose dlog relation is unknown
- ▶ (T_0, T_1) is an ElGamal commitment to t
- ▶ R is the Schnorr commitment
- ▶ c is the Schnorr challenge
- ▶ $ct = (c_0, c_1)$ is the encrypted ATS signature z
- ▶ ρ, ψ are randomizers
- ▶ b_1, \dots, b_n are bits indicating which signers participated

Schnorr TAPS Relation

$$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$$

- ▶ g, h are generators whose dlog relation is unknown
- ▶ (T_0, T_1) is an ElGamal commitment to t
- ▶ R is the Schnorr commitment
- ▶ c is the Schnorr challenge
- ▶ $ct = (c_0, c_1)$ is the encrypted ATS signature z
- ▶ ρ, ψ are randomizers
- ▶ b_1, \dots, b_n are bits indicating which signers participated

Schnorr TAPS Relation

$$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$$

- ▶ g, h are generators whose dlog relation is unknown
- ▶ (T_0, T_1) is an ElGamal commitment to t
- ▶ R is the Schnorr commitment
- ▶ c is the Schnorr challenge
- ▶ $ct = (c_0, c_1)$ is the encrypted ATS signature z
- ▶ ρ, ψ are randomizers
- ▶ b_1, \dots, b_n are bits indicating which signers participated

Schnorr TAPS Relation

$$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$$

- ▶ g, h are generators whose dlog relation is unknown
- ▶ (T_0, T_1) is an ElGamal commitment to t
- ▶ R is the Schnorr commitment
- ▶ c is the Schnorr challenge
- ▶ $ct = (c_0, c_1)$ is the encrypted ATS signature z
- ▶ ρ, ψ are randomizers
- ▶ b_1, \dots, b_n are bits indicating which signers participated

Schnorr TAPS Relation

$$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$$

- ▶ g, h are generators whose dlog relation is unknown
- ▶ (T_0, T_1) is an ElGamal commitment to t
- ▶ R is the Schnorr commitment
- ▶ c is the Schnorr challenge
- ▶ $ct = (c_0, c_1)$ is the encrypted ATS signature z
- ▶ ρ, ψ are randomizers
- ▶ b_1, \dots, b_n are bits indicating which signers participated

Schnorr TAPS Relation

$$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$$

- ▶ g, h are generators whose dlog relation is unknown
- ▶ (T_0, T_1) is an ElGamal commitment to t
- ▶ R is the Schnorr commitment
- ▶ c is the Schnorr challenge
- ▶ $ct = (c_0, c_1)$ is the encrypted ATS signature z
- ▶ ρ, ψ are randomizers
- ▶ b_1, \dots, b_n are bits indicating which signers participated

Schnorr TAPS Relation

$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$ iff

$$(1) \quad g^z = \left[\prod_{i=1}^n (pk_i)^{b_i} \right]^c \cdot R$$

// Proves z is valid for a subset of public keys in PK

$$(2) \quad c_0 = g^\rho \quad \text{and} \quad c_1 = g^z \cdot pk_t^\rho$$

// Proves the ciphertext is a valid encryption to tracer's public key

$$(3) \quad T_0 = g^\psi \quad \text{and} \quad T_1 = g^{\sum_{i=1}^n b_i} \cdot h^\psi$$

// Proves signing quorum contains t signers

$$(4) \quad b_i(1 - b_i) = 0 \quad \text{for } i = 1, \dots, n \quad // \text{ Proves each } b_i \in \{0, 1\}$$

Schnorr TAPS Relation

$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$ iff

$$(1) \quad g^z = \left[\prod_{i=1}^n (pk_i)^{b_i} \right]^c \cdot R$$

// Proves z is valid for a subset of public keys in PK

$$(2) \quad c_0 = g^\rho \quad \text{and} \quad c_1 = g^z \cdot pk_t^\rho$$

// Proves the ciphertext is a valid encryption to tracer's public key

$$(3) \quad T_0 = g^\psi \quad \text{and} \quad T_1 = g^{\sum_{i=1}^n b_i} \cdot h^\psi$$

// Proves signing quorum contains t signers

$$(4) \quad b_i(1 - b_i) = 0 \quad \text{for } i = 1, \dots, n \quad // \text{ Proves each } b_i \in \{0, 1\}$$

Schnorr TAPS Relation

$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$ iff

$$(1) \quad g^z = \left[\prod_{i=1}^n (pk_i)^{b_i} \right]^c \cdot R$$

// Proves z is valid for a subset of public keys in PK

$$(2) \quad c_0 = g^\rho \quad \text{and} \quad c_1 = g^z \cdot pk_t^\rho$$

// Proves the ciphertext is a valid encryption to tracer's public key

$$(3) \quad T_0 = g^\psi \quad \text{and} \quad T_1 = g^{\sum_{i=1}^n b_i} \cdot h^\psi$$

// Proves signing quorum contains t signers

$$(4) \quad b_i(1 - b_i) = 0 \quad \text{for } i = 1, \dots, n \quad // \text{ Proves each } b_i \in \{0, 1\}$$

Schnorr TAPS Relation

$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$ iff

$$(1) \quad g^z = \left[\prod_{i=1}^n (pk_i)^{b_i} \right]^c \cdot R$$

// Proves z is valid for a subset of public keys in PK

$$(2) \quad c_0 = g^\rho \quad \text{and} \quad c_1 = g^z \cdot pk_t^\rho$$

// Proves the ciphertext is a valid encryption to tracer's public key

$$(3) \quad T_0 = g^\psi \quad \text{and} \quad T_1 = g^{\sum_{i=1}^n b_i} \cdot h^\psi$$

// Proves signing quorum contains t signers

$$(4) \quad b_i(1 - b_i) = 0 \quad \text{for } i = 1, \dots, n \quad // \text{ Proves each } b_i \in \{0, 1\}$$

Schnorr TAPS Relation

$\mathcal{R}_S = \left\{ (g, h, pk' = (pk_1, \dots, pk_n), pk_t, T_0, T_1, R, c, ct = (c_0, c_1)) ; (z, \rho, \psi, b_1, \dots, b_n) \right\}$ iff

$$(1) \quad g^z = \left[\prod_{i=1}^n (pk_i)^{b_i} \right]^c \cdot R$$

// Proves z is valid for a subset of public keys in PK

$$(2) \quad c_0 = g^\rho \quad \text{and} \quad c_1 = g^z \cdot pk_t^\rho$$

// Proves the ciphertext is a valid encryption to tracer's public key

$$(3) \quad T_0 = g^\psi \quad \text{and} \quad T_1 = g^{\sum_{i=1}^n b_i} \cdot h^\psi$$

// Proves signing quorum contains t signers

$$(4) \quad b_i(1 - b_i) = 0 \quad \text{for } i = 1, \dots, n \quad // \text{ Proves each } b_i \in \{0, 1\}$$

Performance of Schnorr TAPS

	Public Key Size		Signature Size	
	G	\mathbb{Z}_q	G	\mathbb{Z}_q
Sigma	$\approx 2n$	0	$\approx n$	$\approx 2n$
Bulletproofs	$\approx n + \frac{n}{40}$	0	$\approx \frac{n}{40}$	4

- ▶ Both constructions reduce to discrete logarithm assumptions.

Takeaways

- ▶ TAPS are a new type of threshold signature with both privacy and accountability.
- ▶ We define a generic construction employing an encrypted ATS.
- ▶ We define a Schnorr construction that leverages the structure of Schnorr to simplify the zero-knowledge statement.
- ▶ We define sigma and bulletproofs instantiations of the zero-knowledge argument.

Thank You!

Takeaways

- ▶ TAPS are a new type of threshold signature with both privacy and accountability.
- ▶ We define a generic construction employing an encrypted ATS.
- ▶ We define a Schnorr construction that leverages the structure of Schnorr to simplify the zero-knowledge statement.
- ▶ We define sigma and bulletproofs instantiations of the zero-knowledge argument.

Thank You!

Takeaways

- ▶ TAPS are a new type of threshold signature with both privacy and accountability.
- ▶ We define a generic construction employing an encrypted ATS.
- ▶ We define a Schnorr construction that leverages the structure of Schnorr to simplify the zero-knowledge statement.
- ▶ We define sigma and bulletproofs instantiations of the zero-knowledge argument.

Thank You!

Takeaways

- ▶ TAPS are a new type of threshold signature with both privacy and accountability.
- ▶ We define a generic construction employing an encrypted ATS.
- ▶ We define a Schnorr construction that leverages the structure of Schnorr to simplify the zero-knowledge statement.
- ▶ We define sigma and bulletproofs instantiations of the zero-knowledge argument.

Thank You!

Takeaways

- ▶ TAPS are a new type of threshold signature with both privacy and accountability.
- ▶ We define a generic construction employing an encrypted ATS.
- ▶ We define a Schnorr construction that leverages the structure of Schnorr to simplify the zero-knowledge statement.
- ▶ We define sigma and bulletproofs instantiations of the zero-knowledge argument.

Thank You!