Limitations on Transformations from Composite-Order to Prime-Order Groups: The Case of Round-Optimal Blind Signatures

Sarah Meiklejohn (UC San Diego)
Hovav Shacham (UC San Diego)
David Mandell Freeman (Stanford University)

Elliptic curves: what are they and why do we care?

Bilinear groups are cyclic groups G of some finite order that admit a nondegenerate bilinear map $e: G \times G \to G_T$

- Bilinear: $e(x^a,y) = e(x,y)^a = e(x,y^a)$, nondegenerate: $e(x,y) = 1$ for all $y \Leftrightarrow x = 1$

- Composite order: $|G| = N$ (often use $N = pq$), prime order: $|G| = p$

Elliptic curves: what are they and why do we care?

Bilinear groups are cyclic groups G of some finite order that admit a nondegenerate bilinear map $e: G \times G \rightarrow G_T$

- Bilinear: $e(x^a,y) = e(x,y)^a = e(x,y^a)$, nondegenerate: $e(x,y) = 1$ for all $y \Leftrightarrow x = 1$

- Composite order: $|G| = N$ (often use $N = pq$), prime order: $|G| = p$

Elliptic curves: what are they and why do we care?

Bilinear groups are cyclic groups G of some finite order that admit a nondegenerate bilinear map $e: G \times G \rightarrow G_T$

- Bilinear: $e(x^a,y) = e(x,y)^a = e(x,y^a)$, nondegenerate: $e(x,y) = 1$ for all $y \Leftrightarrow x = 1$

- Composite order: $|G| = N$ (often use $N = pq$), prime order: $|G| = p$

Historically, we use elliptic curves for two main reasons:

- Efficiency: discrete log problem is harder, can use smaller parameters

- Functionality: IBE [BF01], predicate encryption [KSW08], etc.

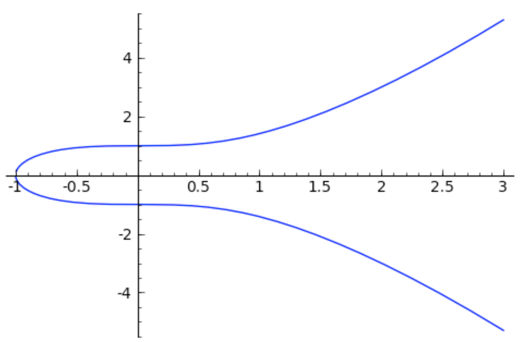
Outline

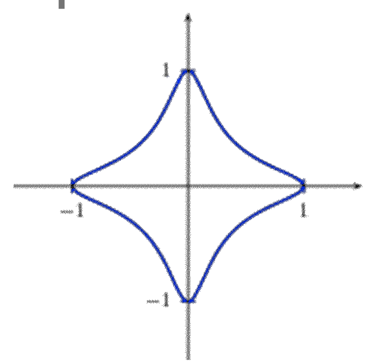
Divide the talk into three main parts:

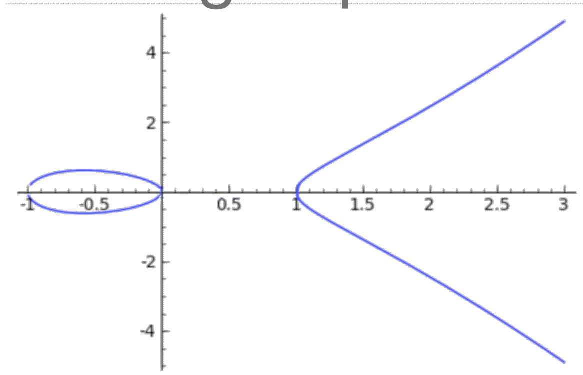
Outline

Divide the talk into three main parts:

- The setting: work in composite-order bilinear groups



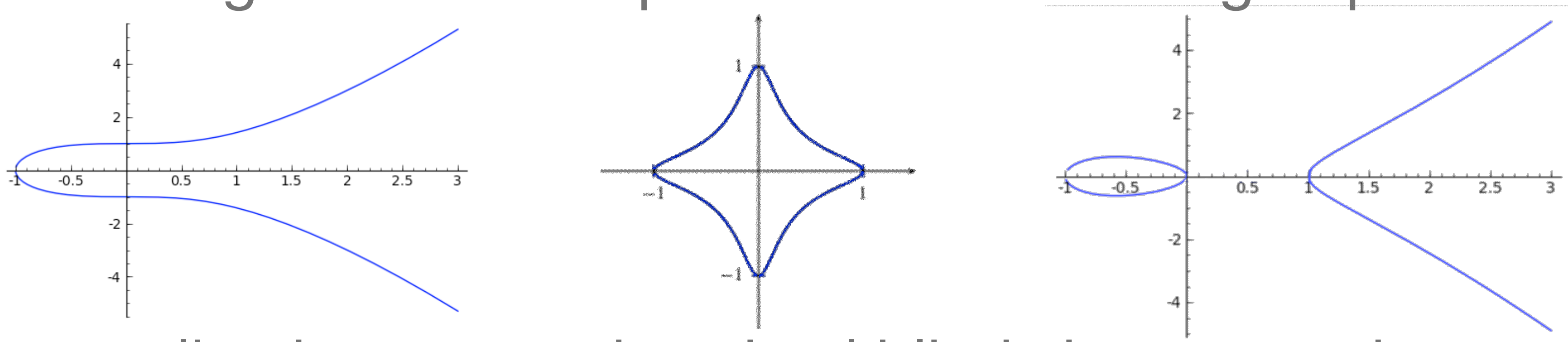




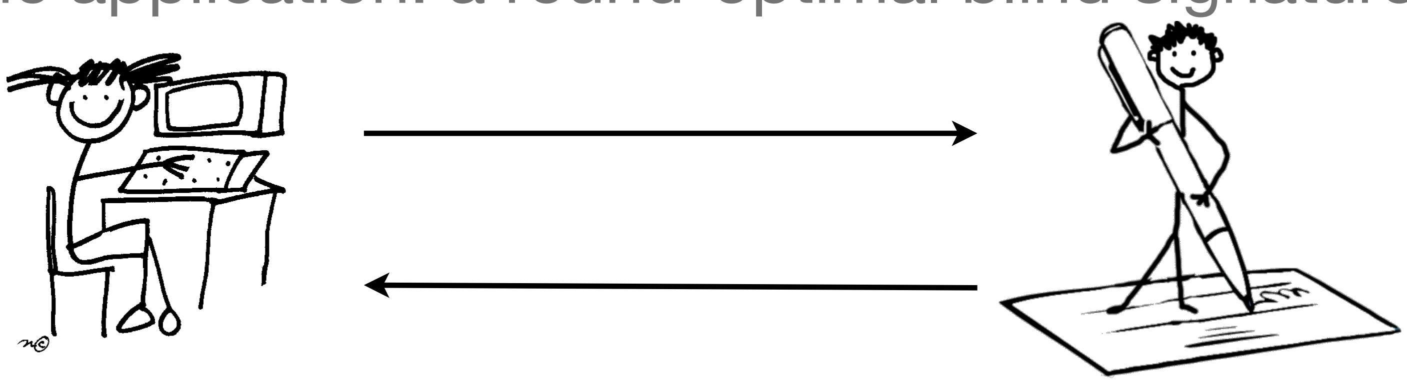
Outline

Divide the talk into three main parts:

- The setting: work in composite-order bilinear groups



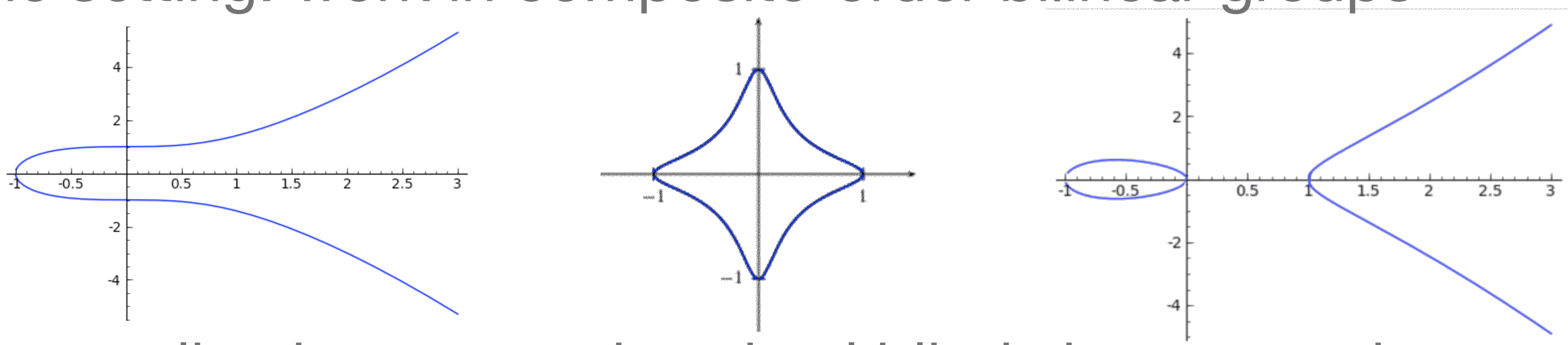
- The application: a round-optimal blind signature scheme



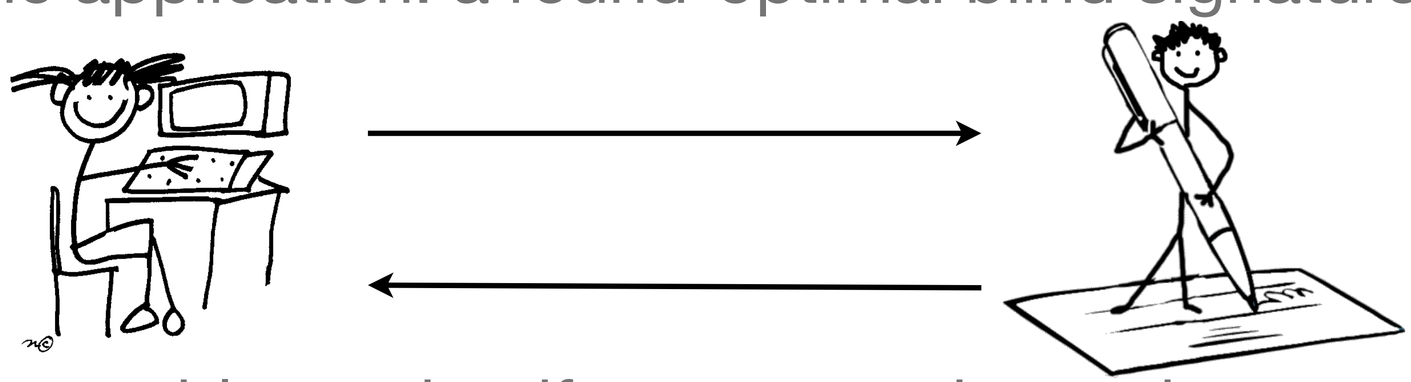
Outline

Divide the talk into three main parts:

- The setting: work in composite-order bilinear groups



- The application: a round-optimal blind signature scheme



- The problem: what if we want to instantiate our scheme in a prime-order setting instead?

The setting: composite-order bilinear groups

- Cyclic groups G and G_T of order $N = pq$, $G = G_p \times G_q$ but p,q are secret
- Bilinear map $e: G \times G \to G_T$
- Often use the subgroup hiding assumption: element of G_q indistinguishable from an element of G
- This setting has proved to be quite useful:

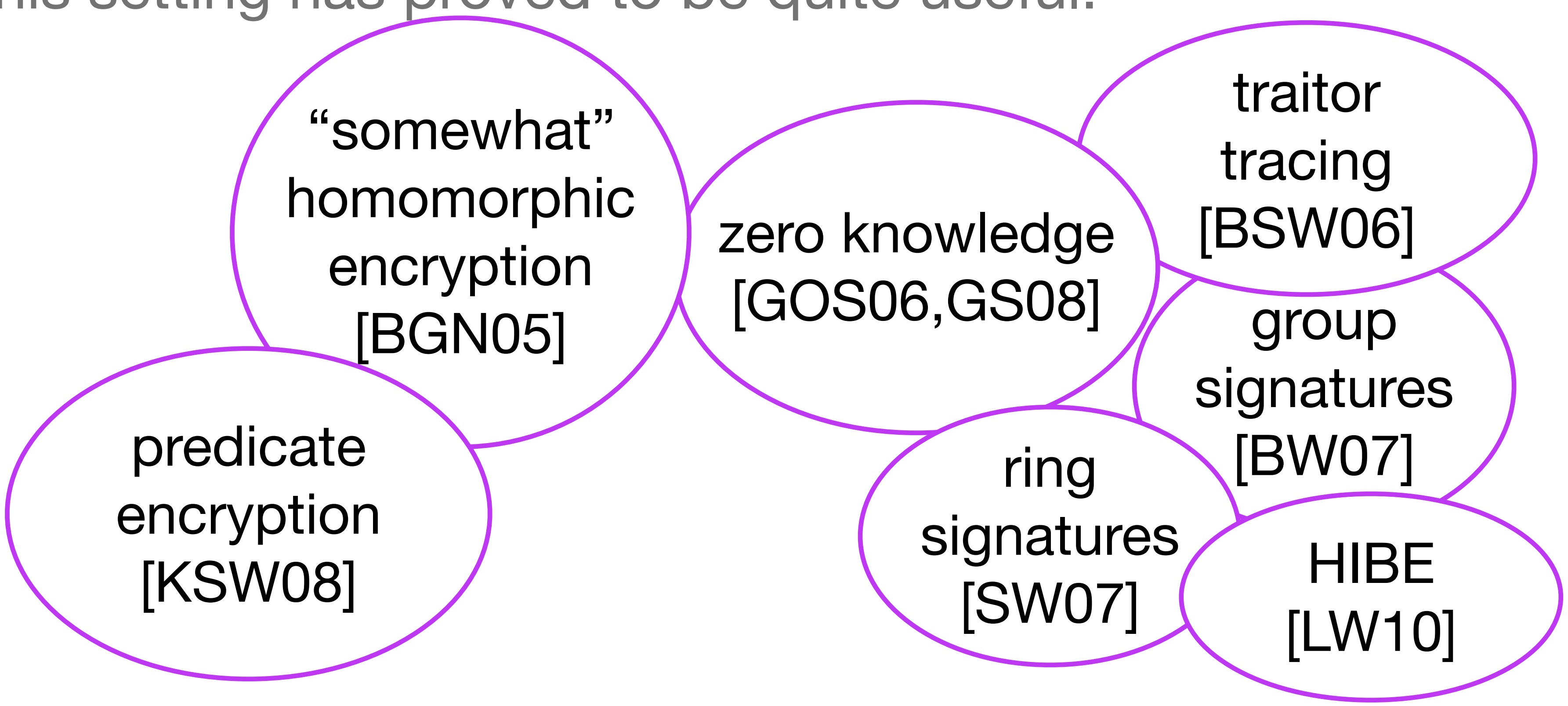
The setting: composite-order bilinear groups

- Cyclic groups G and G_T of order $N = pq$, $G = G_p \times G_q$ but p,q are secret
- Bilinear map $e: G \times G \to G_T$
- Often use the subgroup hiding assumption: element of G_q indistinguishable from an element of G
- This setting has proved to be quite useful:

“somewhat” homomorphic encryption [BGN05]

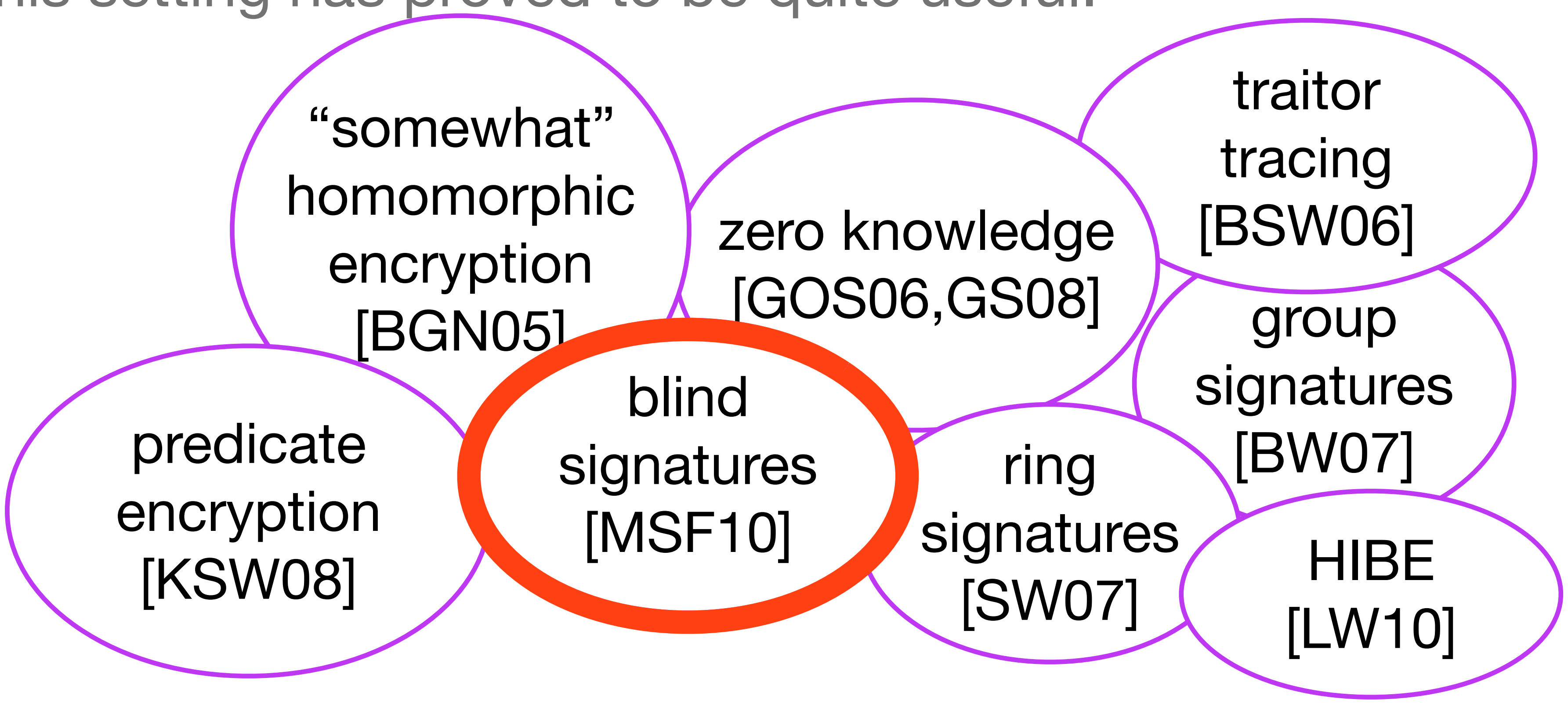
The setting: composite-order bilinear groups

- Cyclic groups G and G_T of order $N = pq$, $G = G_p \times G_q$ but p,q are secret
- Bilinear map $e: G \times G \rightarrow G_T$
- Often use the subgroup hiding assumption: element of G_q indistinguishable from an element of G
- This setting has proved to be quite useful:



The setting: composite-order bilinear groups

- Cyclic groups G and G_T of order $N = pq$, $G = G_p \times G_q$ but p,q are secret
- Bilinear map $e: G \times G \to G_T$
- Often use the subgroup hiding assumption: element of G_q indistinguishable from an element of G
- This setting has proved to be quite useful:



Composite- vs. prime-order groups

Why would we switch to prime-order groups?

Composite- vs. prime-order groups

Why would we switch to prime-order groups?

- Composite-order means bigger: in prime-order groups, can use group of size ~160 bits; in composite-order groups need ~1024 bits (discrete log vs. factoring)

- In addition, there aren't many composite-order curve families (need to use supersingular vs. ordinary curves)

Composite- vs. prime-order groups

Why would we switch to prime-order groups?

- Composite-order means bigger: in prime-order groups, can use group of size ~160 bits; in composite-order groups need ~1024 bits (discrete log vs. factoring)

- In addition, there aren't many composite-order curve families (need to use supersingular vs. ordinary curves)

Previously, people converted schemes in an ad-hoc way [W09,GSW09,LW10]

Freeman [F10] is first to provide a general conversion method

The application: round-optimal blind signatures

Signatures: user obtains a signature σ on a message m from a signer

The application: round-optimal blind signatures

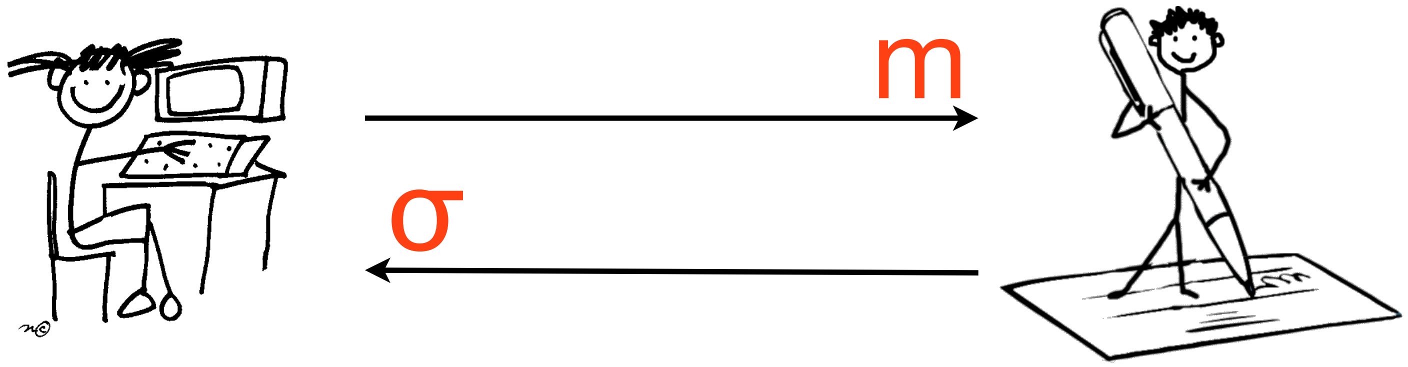
Signatures: user obtains a signature σ on a message m from a signer

The application: round-optimal blind signatures

Signatures: user obtains a signature σ on a message m from a signer

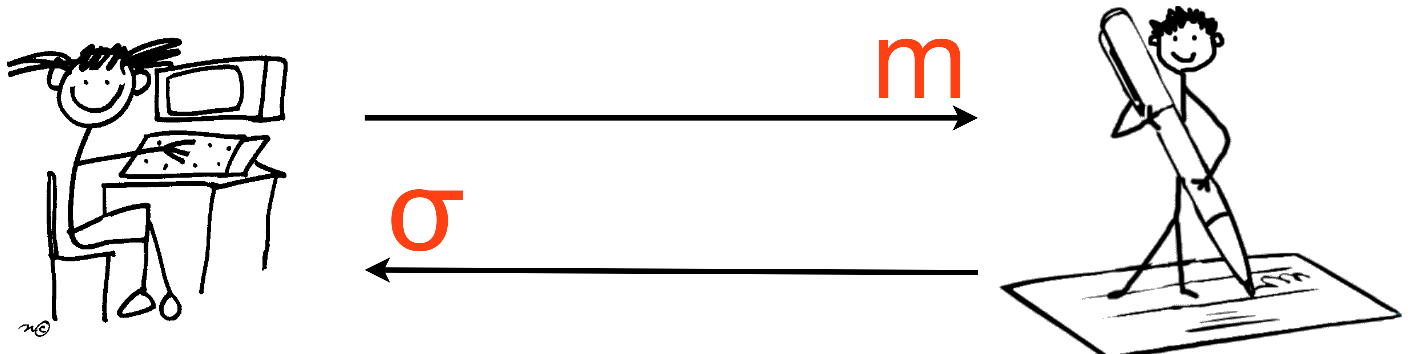
The application: round-optimal blind signatures

Signatures: user obtains a signature σ on a message m from a signer



The application: round-optimal blind signatures

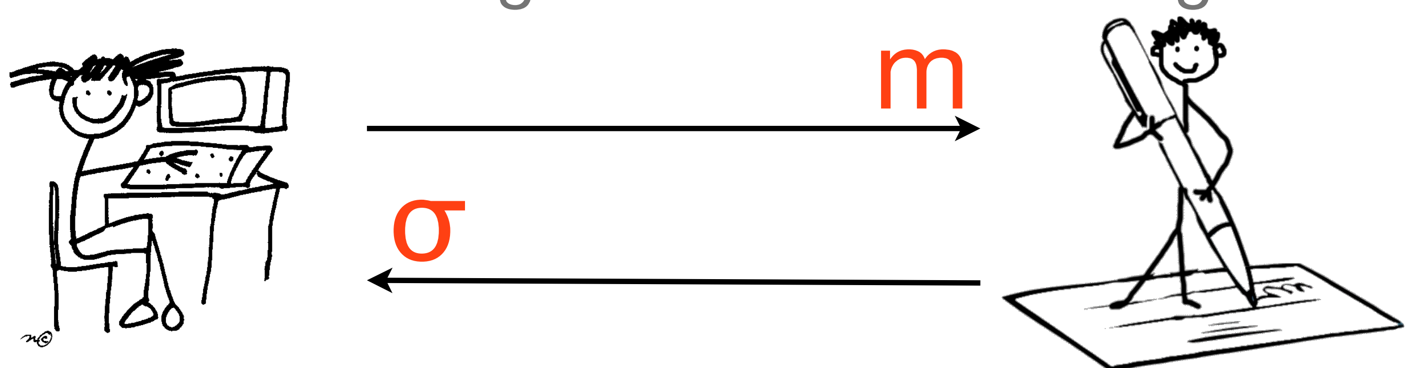
Signatures: user obtains a signature σ on a message m from a signer



In a blind signature scheme [Ch82], user gets this signature without the signer learning which message he signed!

The application: round-optimal blind signatures

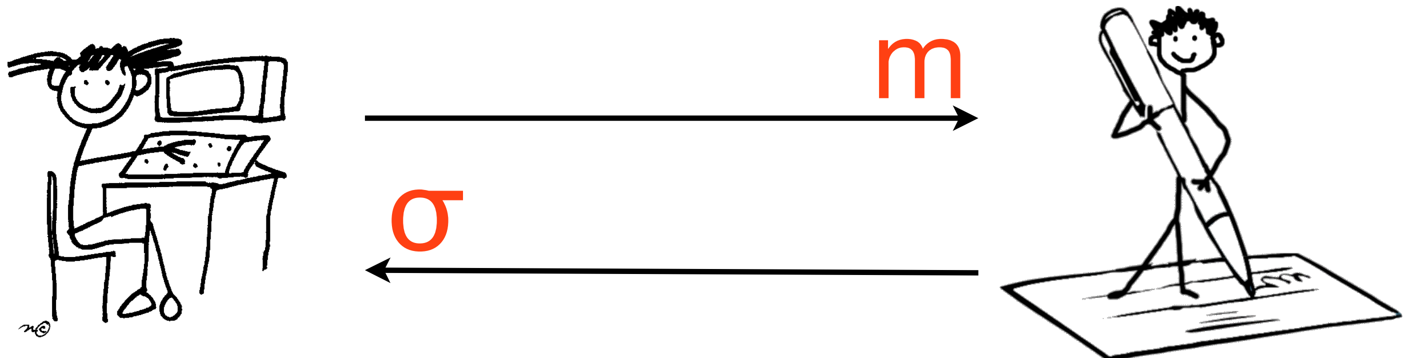
Signatures: user obtains a signature σ on a message m from a signer



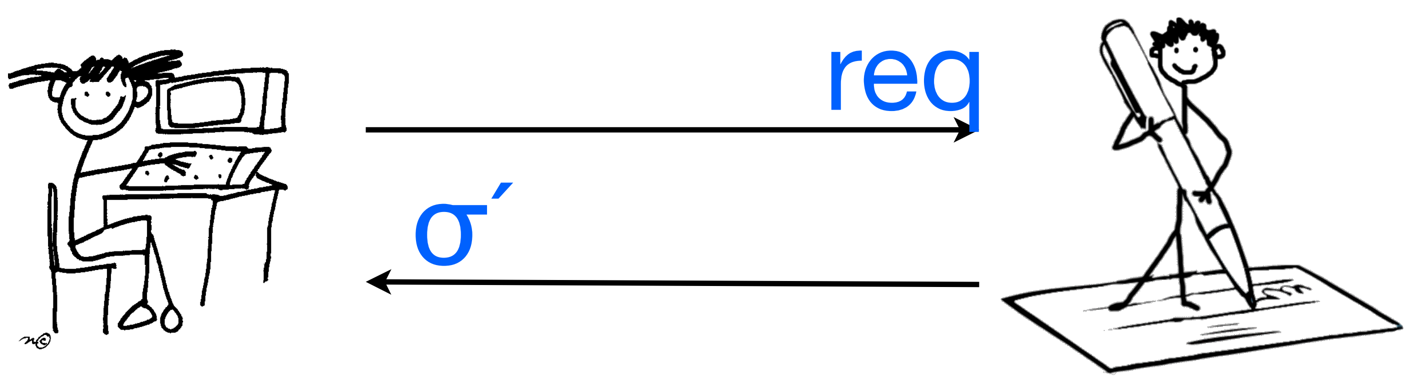
In a blind signature scheme [Ch82], user gets this signature without the signer learning which message he signed!

The application: round-optimal blind signatures

Signatures: user obtains a signature σ on a message m from a signer

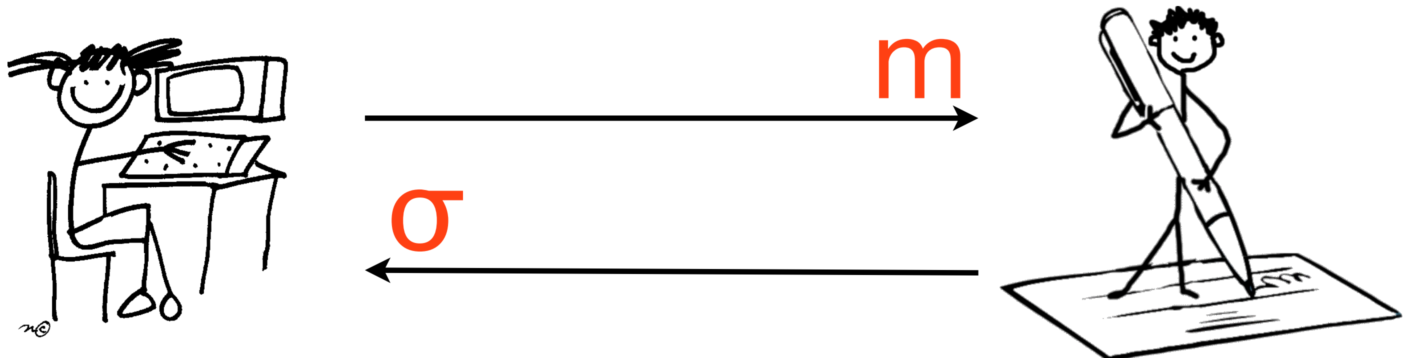


In a blind signature scheme [Ch82], user gets this signature without the signer learning which message he signed!

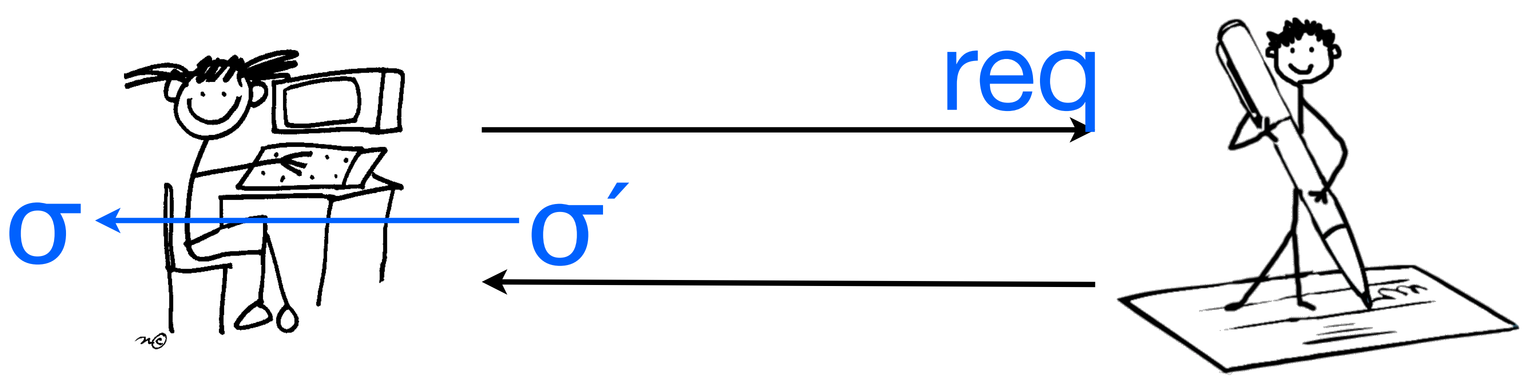


The application: round-optimal blind signatures

Signatures: user obtains a signature σ on a message m from a signer

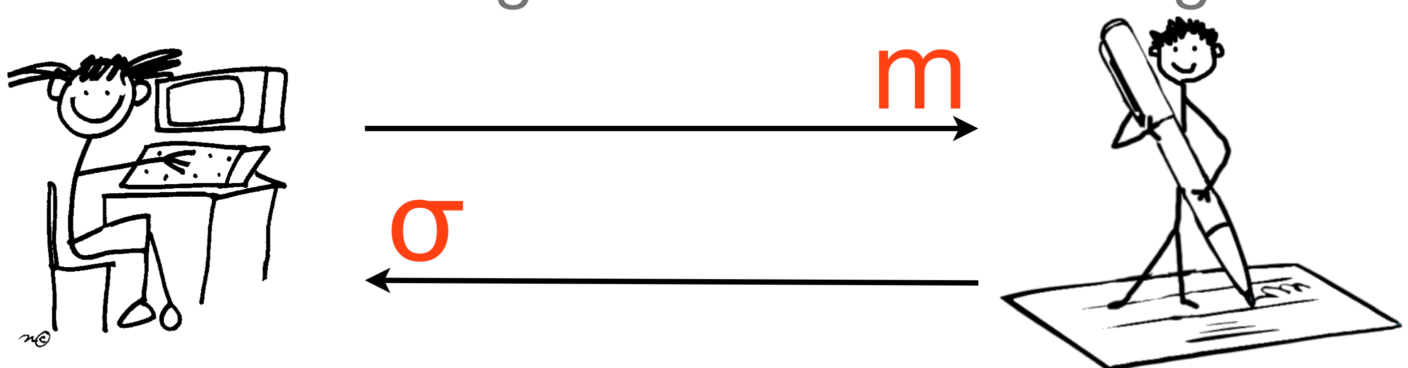


In a blind signature scheme [Ch82], user gets this signature without the signer learning which message he signed!

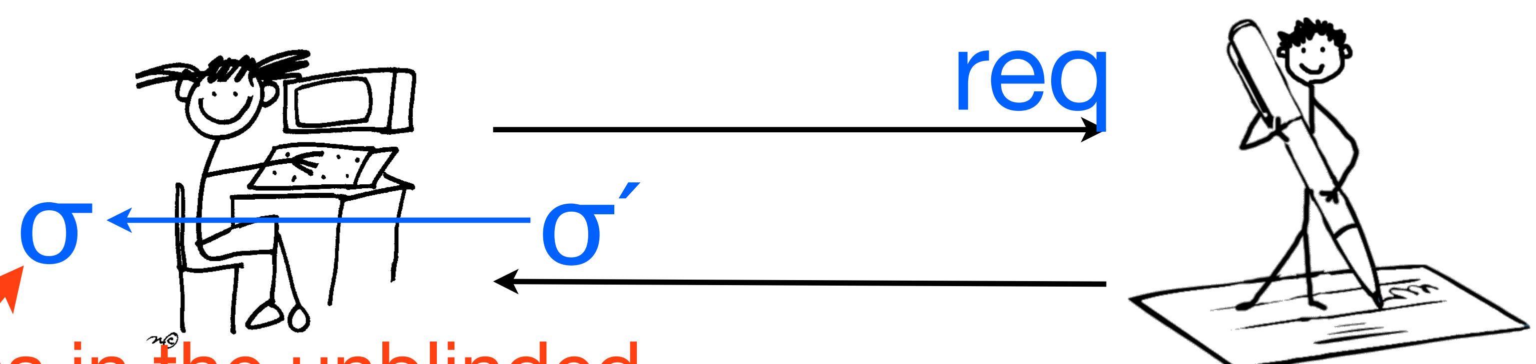


The application: round-optimal blind signatures

Signatures: user obtains a signature σ on a message m from a signer



In a blind signature scheme [Ch82], user gets this signature without the signer learning which message he signed!

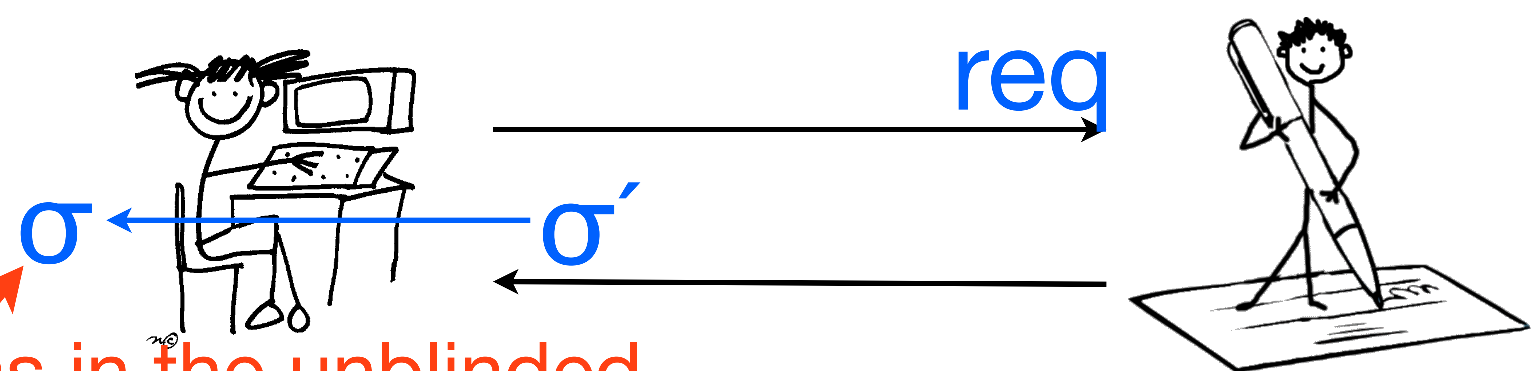


Same σ as in the unblinded case above

The application: round-optimal blind signatures

Signatures: user obtains a signature σ on a message m from a signer

In a blind signature scheme [Ch82], user gets this signature without the signer learning which message he signed!



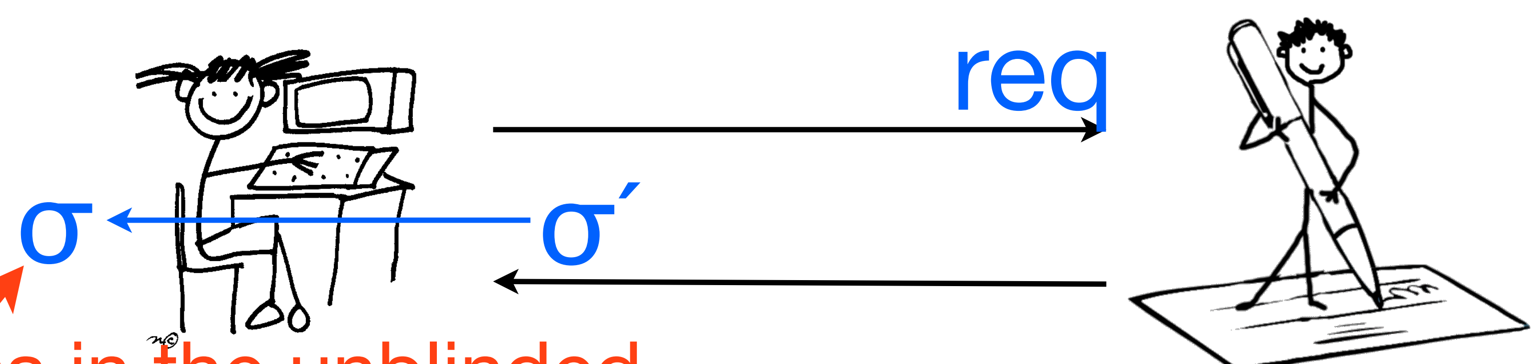
Same σ as in the unblinded case above

Applications: electronic cash [Ch82], anonymous credentials, etc.

The application: round-optimal blind signatures

Signatures: user obtains a signature σ on a message m from a signer

In a blind signature scheme [Ch82], user gets this signature without the signer learning which message he signed!



Same σ as in the unblinded case above

Applications: electronic cash [Ch82], anonymous credentials, etc.

Still a very active research area [O06,F09,AO10,AHO10,AFGHO10,R10]

Our scheme: ideas

Simple construction (inspired by [BW06]): combine Waters signature [W07] with Groth-Sahai zero-knowledge proofs [GS08]

Our scheme: ideas

Simple construction (inspired by [BW06]): combine Waters signature [W07] with Groth-Sahai zero-knowledge proofs [GS08]

Recap of Groth-Sahai setting:

Our scheme: ideas

Simple construction (inspired by [BW06]): combine Waters signature [W07] with Groth-Sahai zero-knowledge proofs [GS08]

Recap of Groth-Sahai setting:

$$e: G \times G \rightarrow G_T$$

Our scheme: ideas

Simple construction (inspired by [BW06]): combine Waters signature [W07] with Groth-Sahai zero-knowledge proofs [GS08]

Recap of Groth-Sahai setting:

$$\begin{array}{l} e\colon G \times G \to G_T \\ \quad \downarrow \tau \\ E\colon B \times B \to B_T \end{array}$$

Our scheme: ideas

Simple construction (inspired by [BW06]): combine Waters signature [W07] with Groth-Sahai zero-knowledge proofs [GS08]

Recap of Groth-Sahai setting:

$$\begin{array}{c} e: G \times G \to G_T \\ \tau \downarrow \\ E: B \times B \to B_T \end{array}$$

- Abstract assumption: $B = B_1 \times B_2$, where B_1 is indistinguishable from B

Our scheme: ideas

Simple construction (inspired by [BW06]): combine Waters signature [W07] with Groth-Sahai zero-knowledge proofs [GS08]

Recap of Groth-Sahai setting:

$$\begin{array}{l} e: G \times G \to G_T \\ \quad \tau \downarrow \\ E: B \times B \to B_T \end{array}$$

- Abstract assumption: $B = B_1 \times B_2$, where B_1 is indistinguishable from B
  - Subgroup hiding: set $B = G = G_p \times G_q$

Our scheme: ideas

Simple construction (inspired by [BW06]): combine Waters signature [W07] with Groth-Sahai zero-knowledge proofs [GS08]

Recap of Groth-Sahai setting:

$$\begin{array}{c} e: G \times G \to G_T \\ \tau \downarrow \\ E: B \times B \to B_T \end{array}$$

- Abstract assumption: $B = B_1 \times B_2$, where B_1 is indistinguishable from B
    - Subgroup hiding: set $B = G = G_p \times G_q$
    - DLIN: rank 2 matrix ~ rank 3 matrix for a 3×3 matrix over F_p

Our scheme: ideas

Simple construction (inspired by [BW06]): combine Waters signature [W07] with Groth-Sahai zero-knowledge proofs [GS08]

Recap of Groth-Sahai setting:

$$\begin{array}{c} e: G \times G \to G_T \\ \tau \downarrow \\ E: B \times B \to B_T \end{array}$$

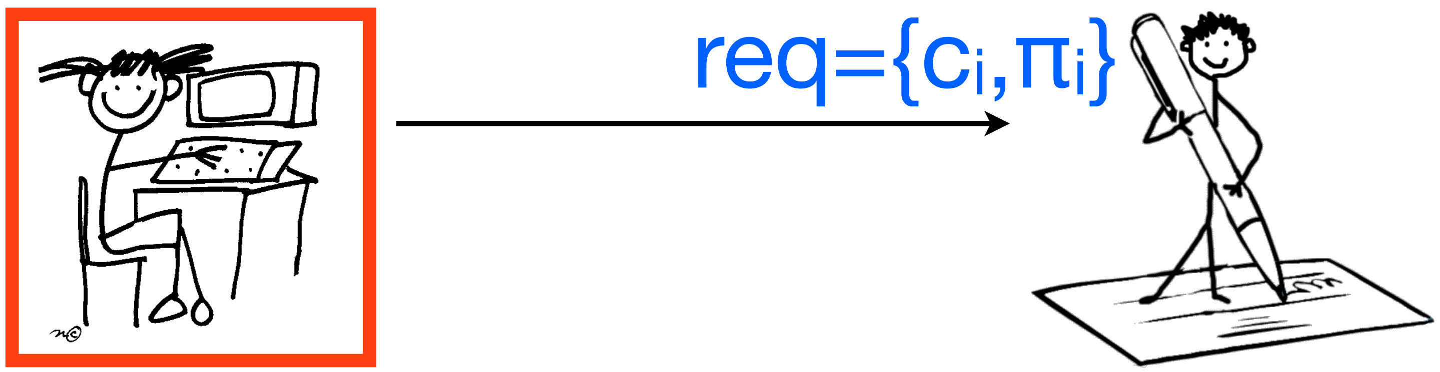
- Abstract assumption: $B = B_1 \times B_2$, where B_1 is indistinguishable from B
  - Subgroup hiding: set $B = G = G_p \times G_q$
  - DLIN: rank 2 matrix ~ rank 3 matrix for a 3×3 matrix over F_p
- Benefits: can use composite- and prime-order settings

Our scheme: sketch

Our scheme: sketch

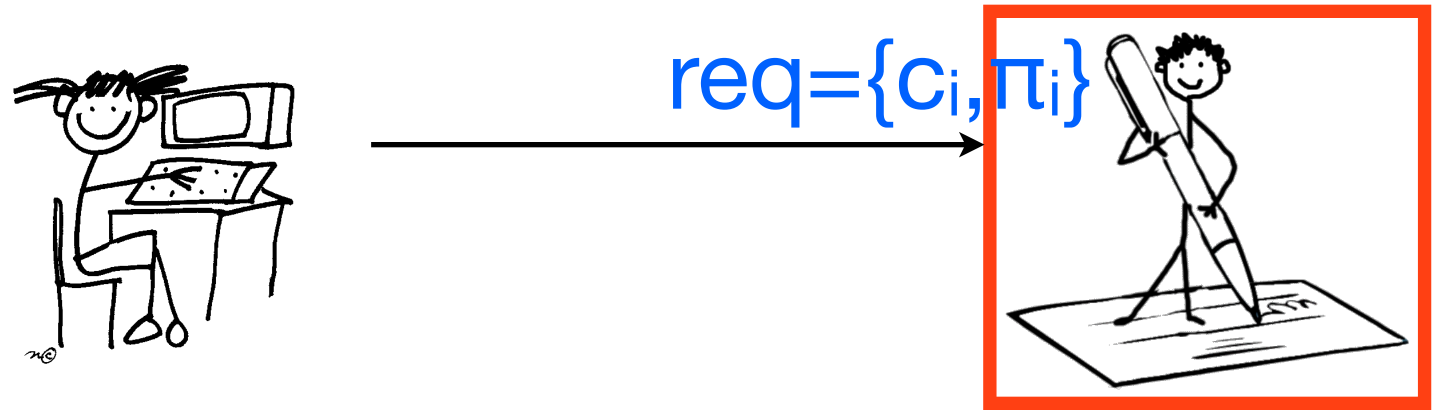
- User: write message bitwise as $m = b_1 \ldots b_n$, compute GS commitment c_i to each bit b_i and GS proof π_i that value in c_i is either 0 or 1

Our scheme: sketch



- User: write message bitwise as $m = b_1...b_n$, compute GS commitment c_i to each bit b_i and GS proof π_i that value in c_i is either 0 or 1

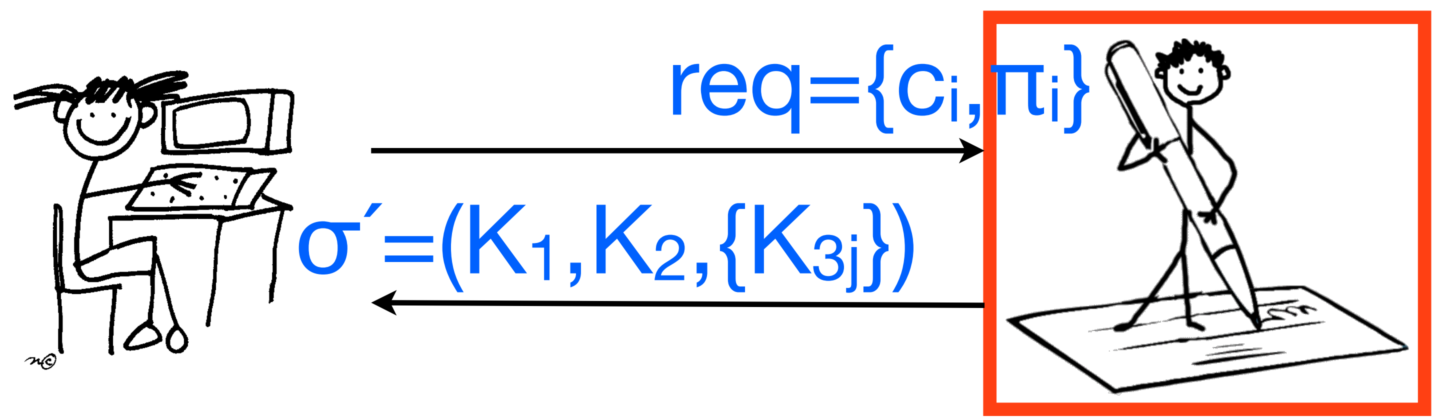
Our scheme: sketch



- User: write message bitwise as $m = b_1 \ldots b_n$, compute GS commitment c_i to each bit b_i and GS proof π_i that value in c_i is either 0 or 1

- Signer: check proof (c_i,π_i) for each i, then compute blind signature $(K_1,K_2,\{K_{3j}\})$

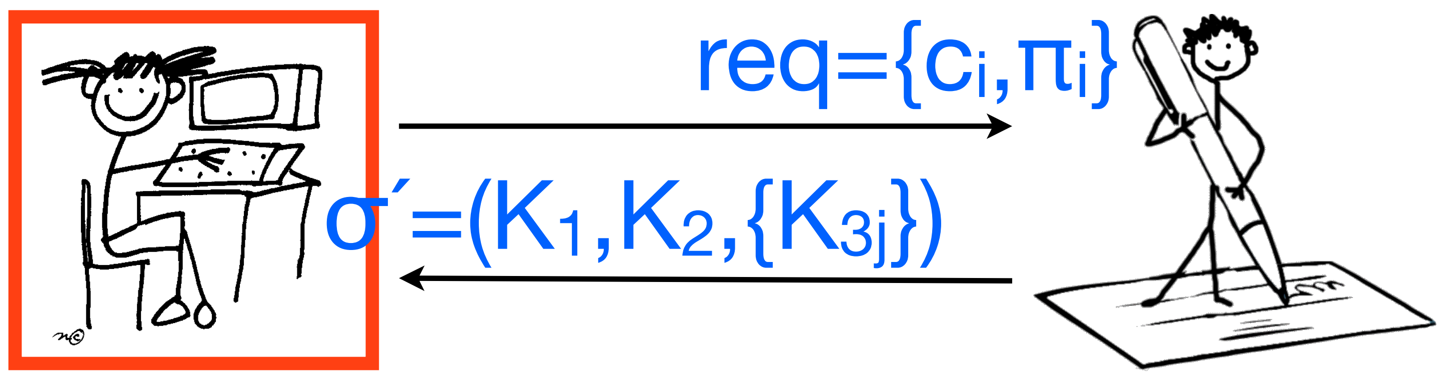
Our scheme: sketch



- User: write message bitwise as $m = b_1...b_n$, compute GS commitment c_i to each bit b_i and GS proof π_i that value in c_i is either 0 or 1

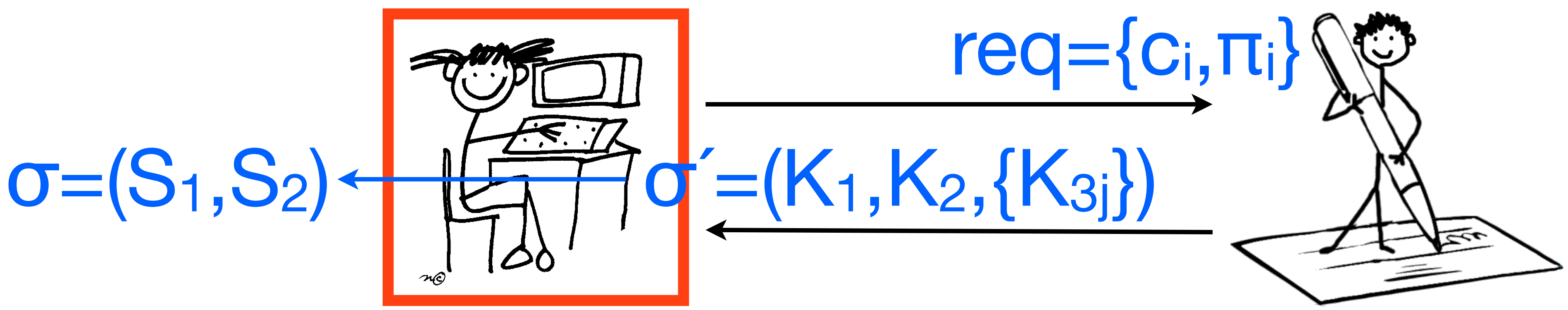
- Signer: check proof (c_i,π_i) for each i, then compute blind signature (K_1,K_2,{K_{3j}})

Our scheme: sketch



- User: write message bitwise as $m = b_1 \ldots b_n$, compute GS commitment c_i to each bit b_i and GS proof π_i that value in c_i is either 0 or 1
- Signer: check proof (c_i,π_i) for each i, then compute blind signature $(K_1,K_2,\{K_{3j}\})$
- User: check blind signature was formed properly, then unblind it using randomness from the commitments to get Waters signature (S_1,S_2)

Our scheme: sketch

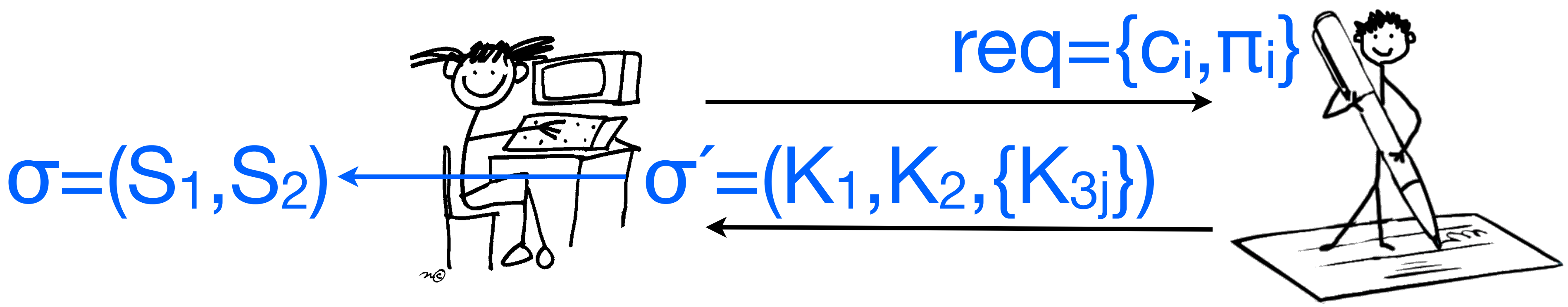


- User: write message bitwise as $m = b_1...b_n$, compute GS commitment c_i to each bit b_i and GS proof π_i that value in c_i is either 0 or 1

- Signer: check proof (c_i,π_i) for each i, then compute blind signature $(K_1,K_2,\{K_{3j}\})$

- User: check blind signature was formed properly, then unblind it using randomness from the commitments to get Waters signature (S_1,S_2)

Our scheme: sketch



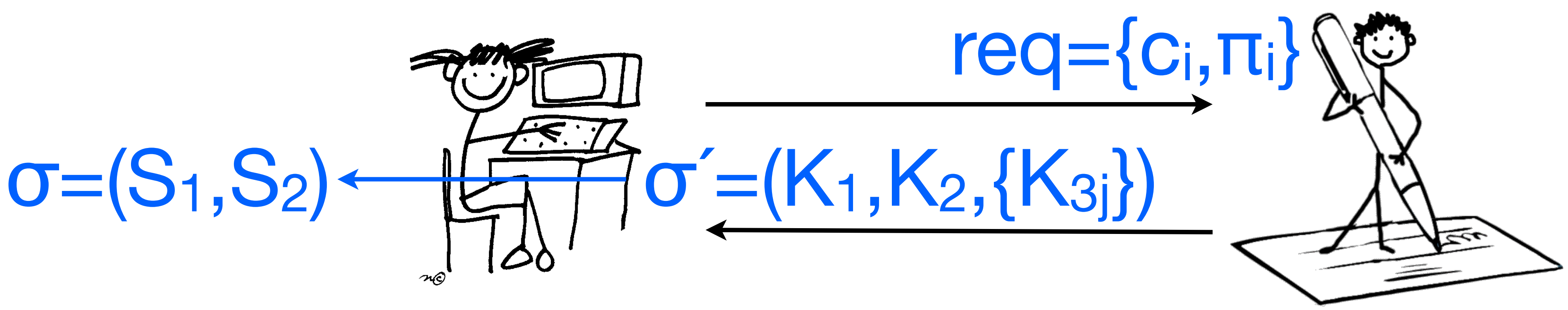
- User: write message bitwise as $m = b_1 \ldots b_n$, compute GS commitment c_i to each bit b_i and GS proof π_i that value in c_i is either 0 or 1

Request is a bit long, but...

- Signer: check proof (c_i,π_i) for each i, then compute blind signature $(K_1,K_2,\{K_{3j}\})$

- User: check blind signature was formed properly, then unblind it using randomness from the commitments to get Waters signature (S_1,S_2)

Our scheme: sketch



- User: write message bitwise as $m = b_1 \ldots b_n$, compute GS commitment c_i to each bit b_i and GS proof π_i that value in c_i is either 0 or 1

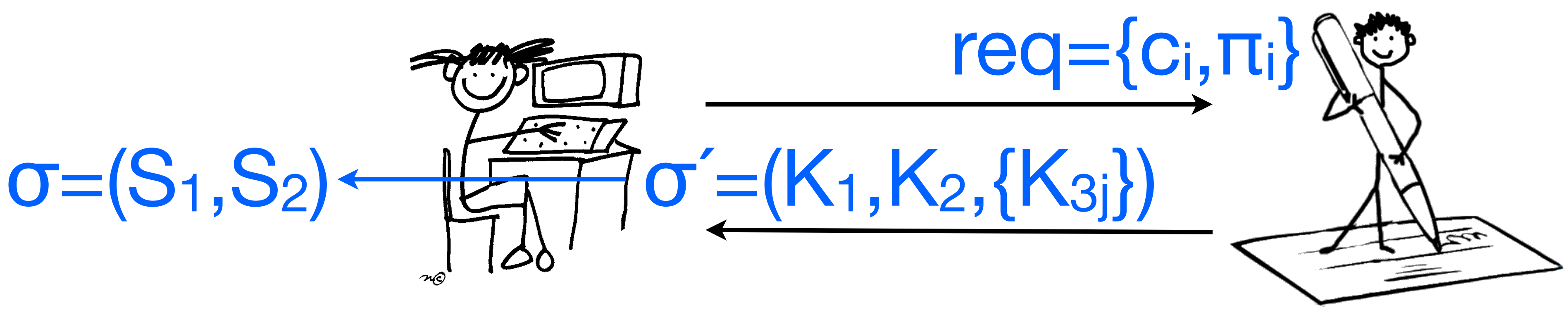
Request is a bit long, but...

- Signer: check proof (c_i,π_i) for each i, then compute blind signature $(K_1,K_2,\{K_{3j}\})$

- User: check blind signature was formed properly, then unblind it using randomness from the commitments to get Waters signature (S_1,S_2)

...blind signature is short (j=1,2,or 3), and...

Our scheme: sketch



- User: write message bitwise as $m = b_1 \ldots b_n$, compute GS commitment c_i to each bit b_i and GS proof π_i that value in c_i is either 0 or 1

Request is a bit long, but...

- Signer: check proof (c_i,π_i) for each i, then compute blind signature $(K_1,K_2,\{K_{3j}\})$

- User: check blind signature was formed properly, then unblind it using randomness from the commitments to get Waters signature (S_1,S_2)

...signature obtained is short as well!

...blind signature is short (j=1,2,or 3), and...

Our scheme: security

Can prove the following security theorem:

- Under the subgroup hiding and CDH assumptions, our blind signature scheme is one-more unforgeable and blind (using the standard definitions [JLO97]).

Our scheme: security

Can prove the following security theorem:

- Under the subgroup hiding and CDH assumptions, our blind signature scheme is one-more unforgeable and blind (using the standard definitions [JLO97]).

Can we prove a more abstract theorem?

Our scheme: security

Can prove the following security theorem:

- Under the subgroup hiding and CDH assumptions, our blind signature scheme is one-more unforgeable and blind (using the standard definitions [JLO97]).

Can we prove a more abstract theorem?

- Blindness requires only the abstract assumption,...
- ... but one-more unforgeability requires more.

Projecting and cancelling

Security proof relies on two properties: projecting and cancelling

Projecting and cancelling

Security proof relies on two properties: projecting and cancelling

For projecting, we have:

- decomposition $B = B_1 \times B_2$
- map $\pi: B \rightarrow B_2$ s.t. $\pi(b=b_1 * b_2) = b_2$
- map π_T s.t. $\pi_T(E(a,b)) = E(\pi(a),\pi(b))$

Projecting and cancelling

Security proof relies on two properties: projecting and cancelling

For projecting, we have:

- decomposition $B = B_1 \times B_2$
- map $\pi: B \to B_2$ s.t. $\pi(b=b_1{*}b_2) = b_2$
- map π_T s.t. $\pi_T(E(a,b)) = E(\pi(a),\pi(b))$

For cancelling, we have:

- decomposition $B = B_1 \times B_2$ such that $E(a,b) = 1$ for all a in B_1, b in B_2

Projecting and cancelling

Security proof relies on two properties: projecting and cancelling

For projecting, we have:

- decomposition $B = B_1 \times B_2$
- map $\pi: B \to B_2$ s.t. $\pi(b=b_1{*}b_2) = b_2$
- map π_T s.t. $\pi_T(E(a,b)) = E(\pi(a),\pi(b))$

For cancelling, we have:

- decomposition $B = B_1 \times B_2$ such that $E(a,b) = 1$ for all a in B_1, b in B_2

In composite-order groups:

$B = G = G_p \times G_q$

Projecting: $\pi(x) = x^\lambda$ for λ s.t.

$\lambda = 0 \bmod p$

$\lambda = 1 \bmod q$

Then $\pi(g) = \pi(g_p * g_q) = (g^q * g^p)^\lambda = g_q$

Cancelling:

$E(g_p,g_q) = E(g^q,g^p) = E(g,g)^{pq} = E(g,g)^N = 1$

Projecting and cancelling

Security proof relies on two properties: projecting and cancelling

For projecting, we have:

- decomposition $B = B_1 \times B_2$
- map π: $B \rightarrow B_2$ s.t. $\pi(b=b_1*b_2) = b_2$
- map π_T s.t. $\pi_T(E(a,b)) = E(\pi(a),\pi(b))$

For cancelling, we have:

- decomposition $B = B_1 \times B_2$ such that $E(a,b) = 1$ for all a in B_1, b in B_2

In composite-order groups:

$B = G = G_p \times G_q$

Projecting: $\pi(x) = x^\lambda$ for λ s.t.
$\lambda = 0 \bmod p$
$\lambda = 1 \bmod q$
Then $\pi(g) = \pi(g_p * g_q) = (g^q * g^p)^\lambda = g_q$

Cancelling:
$E(g_p,g_q) = E(g^q,g^p) = E(g,g)^{pq} = E(g,g)^N = 1$

Freeman [F10] provides generic transformation to prime-order groups for schemes in composite-order groups that require either of these two properties

The problem: what if we want both properties?

This turns out to be very tricky!

The problem: what if we want both properties?

This turns out to be very tricky!

We want to prove the following theorem:

- If we use the DLIN assumption for the indistinguishability of B_1 and B and E is cancelling, then E cannot be projecting.

The problem: what if we want both properties?

This turns out to be very tricky!

We want to prove the following theorem:

- If we use the DLIN assumption for the indistinguishability of B_1 and B and E is cancelling, then E cannot be projecting.

Break it up into two lemmas:

- Cancelling shrinks the target space: If we use the DLIN assumption for the indistinguishability of B_1 and B and E is cancelling, then $|E(B,B)| = p$.

- Can't project with small target: If $|E(B,B)| = p$ then E cannot be projecting.

The problem: what if we want both properties?

This turns out to be very tricky!

We want to prove the following theorem:

- If we use the DLIN assumption for the indistinguishability of B_1 and B and E is cancelling, then E cannot be projecting.

Break it up into two lemmas:

- Cancelling shrinks the target space: If we use the DLIN assumption for the indistinguishability of B_1 and B and E is cancelling, then $|E(B,B)| = p$.

✓ Can't project with small target: If $|E(B,B)| = p$ then E cannot be projecting.

The problem: what if we want both properties?

We can prove the following theorem:

- If we use the DLIN assumption* for the indistinguishability of B_1 and B and E is cancelling, then E cannot be projecting with overwhelming probability.

Break it up into two lemmas:

- Let $E: B \times B \to B_T$ be a nondegenerate pairing that is independent of the decomposition $B = B_1 \times B_2$. Then if $B = G^3$, B_1 is a uniformly random rank-2 submodule of B, and E is cancelling, then $|E(B,B)| = p$ with overwhelming probability.

✓ Can't project with small target: If $|E(B,B)| = p$ then E cannot be projecting.

The problem: what if we want both properties?

We can prove the following theorem:

- If we use the DLIN assumption* for the indistinguishability of B_1 and B and E is cancelling, then E cannot be projecting with overwhelming probability.

Break it up into two lemmas:

E is public, if dependent on B_1 could reveal information to help to distinguish it from B

- Let $E: B \times B \to B_T$ be a nondegenerate pairing that is independent of the decomposition $B = B_1 \times B_2$. Then if $B = G^3$, B_1 is a uniformly random rank-2 submodule of B, and E is cancelling, then $|E(B,B)| = p$ with overwhelming probability.

✓ Can't project with small target: If $|E(B,B)| = p$ then E cannot be projecting.

The problem: what if we want both properties?

We can prove the following theorem:

- If we use the DLIN assumption* for the indistinguishability of B_1 and B and E is cancelling, then E cannot be projecting with overwhelming probability.

Break it up into two lemmas:

E is public, if dependent on B_1 could reveal information to help to distinguish it from B

- Let $E: B \times B \rightarrow B_T$ be a nondegenerate pairing that is independent of the decomposition $B = B_1 \times B_2$. Then if $B = G^3$, B_1 is a uniformly random rank-2 submodule of B, and E is cancelling, then $|E(B,B)| = p$ with overwhelming probability.

If B_1 is *not* random, can't be sure indistinguishability still holds

Can't project with small target: If $|E(B,B)| = p$ then E cannot be projecting.

Conclusions

Showed that if we want projecting and cancelling, generic transformations from composite- to prime-order groups fail

- Can't use DLIN (more generally, k-Linear [HK07,S07])
- This suggests possible functionality gap

Conclusions

Showed that if we want projecting and cancelling, generic transformations from composite- to prime-order groups fail

- Can't use DLIN (more generally, k-Linear [HK07,S07])
- This suggests possible functionality gap

Constructed a round-optimal blind signature scheme

- First efficient scheme using 'mild' assumptions (non-interactive, static), even including ones in the random oracle model
- Signature scheme demonstrates potential need for both properties

Open problems

Positive:

Negative:

Open problems

Positive:

- Construct a projecting and cancelling pairing in prime-order groups

Negative:

Open problems

Positive:

- Construct a projecting and cancelling pairing in prime-order groups

Negative:

- Prove there can be no projecting and cancelling pairing in prime-order groups

Open problems

Positive:

- Construct a projecting and cancelling pairing in prime-order groups

- Prove our scheme secure in prime-order groups

Negative:

- Prove there can be no projecting and cancelling pairing in prime-order groups

Open problems

Positive:

- Construct a projecting and cancelling pairing in prime-order groups
- Prove our scheme secure in prime-order groups

Negative:

- Prove there can be no projecting and cancelling pairing in prime-order groups
- Show our scheme is insecure in prime-order groups

Open problems

Positive:

- Construct a projecting and cancelling pairing in prime-order groups
- Prove our scheme secure in prime-order groups
- Show another general conversion from composite- to prime-order groups

Negative:

- Prove there can be no projecting and cancelling pairing in prime-order groups
- Show our scheme is insecure in prime-order groups

Open problems

Positive:

- Construct a projecting and cancelling pairing in prime-order groups
- Prove our scheme secure in prime-order groups
- Show another general conversion from composite- to prime-order groups

Negative:

- Prove there can be no projecting and cancelling pairing in prime-order groups
- Show our scheme is insecure in prime-order groups
- Prove that some other properties cannot be achieved in prime-order groups

Open problems

Positive:

- Construct a projecting and cancelling pairing in prime-order groups
- Prove our sch[illegible]
- Show another [illegible]e-order groups

Any questions?

Negative:

- Prove there ca[illegible] prime-order groups
- Show our scheme is insecure in prime-order groups
- Prove that some other properties cannot be achieved in prime-order groups