Double Feature:

Secure Two-Party Threshold ECDSA from ECDSA Assumptions

Threshold ECDSA from ECDSA Assumptions: the Multiparty Case

Jack Doerner · Yashvanth Kondi · Eysa Lee · abhi shelat
14 Papers!

[GGN16] [Lin17] [BGG17] [DKLs18] [LNR18] [GG18] [DKLs19] [CCLST19] [DOKSS19] [CCLST20] [CMP20] [GKSS20] [DJNPOØ20] [GG20]

2016 2020
14 Papers!

After 7 followups, we still stand out
Securing DNSSEC Keys
via Threshold ECDSA
From Generic MPC

[DKLs18] [DKLs19]

OT-Based
Preproc All But Last Msg

Threshold ECDSA From ECDSA Assumptions

OT-Based
Preproc All But Last Msg

[DKLs18]

OT-Based
Preproc All But Last Msg

[DKLs19]
Securing DNSSEC Keys via Threshold ECDSA From Generic MPC

[DKLs18] [DKLs19]

OT-Based
Preproc All But Last Msg
No EC Abstraction

OT-Based
Preproc All But Last Msg
Nice EC Abstraction

Threshold ECDSA From ECDSA Assumptions

OT-Based
Preproc All But Last Msg
No EC Abstraction
Securing DNSSEC Keys via Threshold ECDSA From Generic MPC

[DKLs18] [DKLs19] [DOKSS19]

OT-Based
Preproc All But Last Msg
No EC Abstraction
2 Msgs for 2 Parties
Many Optimizations

OT-Based
Preproc All But Last Msg
Nice EC Abstraction
Securing DNSSEC Keys via Threshold ECDSA From Generic MPC

[DKLs18] [DKLs19]

OT-Based
Preproc All But Last Msg
No EC Abstraction
2 Msgs for 2 Parties
Many Optimizations
Better 2P Perf

[DKLs18] [DKLs19]

OT-Based
Preproc All But Last Msg
Nice EC Abstraction

Good 2P Perf
Threshold ECDSA
From ECDSA Assumptions

[DKLs18] [DKLs19]

OT-Based
Preproc All But Last Msg
No EC Abstraction
2 Msgs for 2 Parties
Many Optimizations
Better 2P Perf
More Complex Proof

Securing DNSSEC Keys
via Threshold ECDSA
From Generic MPC

[DOKSS19]

OT-Based
Preproc All But Last Msg
Nice EC Abstraction

Good 2P Perf
Simpler Proof
OT vs HE
OT vs HE

Paillier + ZK:

[DKLs18] [GGN16] [Lin17] [BGG17] [LNR18] [GG18] [CMP20] [GKSS20] [GG20]

[DKLs19] [CCLST19] [DOKSS19] [CCLST20]
OT vs HE

CG + HPS: [CCLST19] [CCLST20] [GGN16] [Lin17] [BGG17] [LNR18] [GG18] [CMP20] [GKSS20] [GG20]

Paillier + ZK: [DKLS18] [DKLS19] [DOKSS19] [DJNPØ20]
OT vs HE

CG + HPS: [CCLST19] [CCLST20] [GGN16] [Lin17] [BGG17] [LNR18] [GG18] [CMP20] [GKSS20] [GG20]

Paillier + ZK: [DKLs18] [DKLs19] [DOKSS19]

OT: [DKLs18] [DKLs19] [DOKSS19]
Paillier + ZK  OT
Paillier + ZK

Low Communication

OT

High Communication

No Extra Assumptions

Low Computation

No ZK
Paillier + ZK

- Low Communication
- Extra Assumptions

OT

- High Communication
- No Extra Assumptions
- Low Computation
- No ZK
Paillier + ZK

- Low Communication
- Extra Assumptions
- Very High Computation

OT
Paillier + ZK

- Low Communication
- Extra Assumptions
- Very High Computation
- NIZK over Crypto

OT
Paillier + ZK

- Low Communication
- Extra Assumptions
- Very High Computation
- NIZK over Crypto

OT

- High Communication
- Native Assumptions
- Low Computation
- No ZK
Not so bad, actually

OT

High Communication
Native Assumptions
Low Computation
No ZK
Example 1: Mobile Wallet

Multiplier: OT-based
Parties: 4
Curve: 256-bit

2 Mbits
sent per party
Example 1: Mobile Wallet

Multiplier: OT-based
Parties: 4
Curve: 256-bit

2 Mbits
sent per party
Example 1: Mobile Wallet

Multiplier: OT-based
Parties: 4
Curve: 256-bit

2 Mbits
sent per party
Example 1: Mobile Wallet

Multiplier: OT-based
Parties: 4
Curve: 256-bit

2 Mbits
sent per party

Rank: 25
Avg. Upload: 7.5 Mbps

source: opensignal
Example 1: Mobile Wallet

Multiplier: OT-based
Parties: 4
Curve: 256-bit

2 Mbits
sent per party

Rank: 25
Avg. Upload: 7.5 Mbps

Rank: 86
Avg. Upload: 2.7 Mbps

source: opensignal
Example 1: Mobile Wallet

Multiplier: OT-based
Parties: 4
Curve: 256-bit

2 Mbits
sent per party

Rank: 25
Avg. Upload: 7.5 Mbps
Signing Time: ~1/3 sec

Rank: 86
Avg. Upload: 2.7 Mbps
Signing Time: ~1 sec

source: opensignal
Example 1: Mobile Wallet

Multiplier: OT-based
Parties: 4
Curve: 256-bit
2 Mbits sent per party

Rank: 25
Avg. Upload: 7.5 Mbps
Signing Time: ~1/3 sec

Rank: 86
Avg. Upload: 2.7 Mbps
Signing Time: ~1 sec

Similar to computation time for Paillier on powerful hardware!

source: opensignal
On the Other Hand

Paillier + ZK

OT
Example 2: Datacenter Signing

How much bandwidth to be CPU bound?
(including preprocessing)

2 Parties
~250 sigs/second

256 Parties
~3 sigs/second

using GCP n1-highcpu nodes
Example 2: Datacenter Signing

How much bandwidth to be CPU bound?
(including preprocessing)

2 Parties
~250 sigs/second

Each party sends:
~700 Kbits per sig

256 Parties
~3 sigs/second

Each party sends:
~185 Mbits per sig

using GCP n1-highcpu nodes
Example 2: Datacenter Signing

How much bandwidth to be CPU bound? (including preprocessing)

2 Parties
~250 sigs/second
Each party sends:
~700 Kbits per sig
Bandwidth required:
~180 Mbps symmetric

256 Parties
~3 sigs/second
Each party sends:
~185 Mbits per sig
Bandwidth required:
~555 Mbps symmetric

using GCP n1-highcpu nodes
Summary

Bandwidth isn’t always the bottleneck or the most important cost factor

Guide concrete optimization by studying real use-cases

We ❤ OT
Our Protocols

UC Sec From CDH in the ROM
OT-Based
No ZK in Signing
One “Online” Msg
Const or Log Round Preprocessing
2 Msgs for 2 Parties

Secure Two-Party Threshold ECDSA from ECDSA Assumptions
http://ia.cr/2018/499

Threshold ECDSA from ECDSA Assumptions: the Multiparty Case
http://ia.cr/2019/523