Secure Multiparty Computation and Applications

Presenter: Steve Lu, CEO
Stealth Software Technologies, Inc.

NIST MPTS 2023 Workshop

Sept 26, 2023

STEALTH SOFTWARE TECHNOLOGIES

Stealth Team – About Us

Our Mission

To help organizations maximize the utility of their data while mitigating security and privacy risks through applied cryptography and cybersecurity.

HQ in Los Angeles, co-founded by UCLA Distinguished Professor Rafail Ostrovsky. We additionally work with consultants who are world-renowned cryptographers.

Founded in 2006. We have competitively won over a dozen major contracts from leading agencies in cryptography.

Multiple breakthroughs in cryptography, matured for deployment, including privacy-preserving computation, analysis, and storage.

Centralized Systems

Registry

Certificate Authority

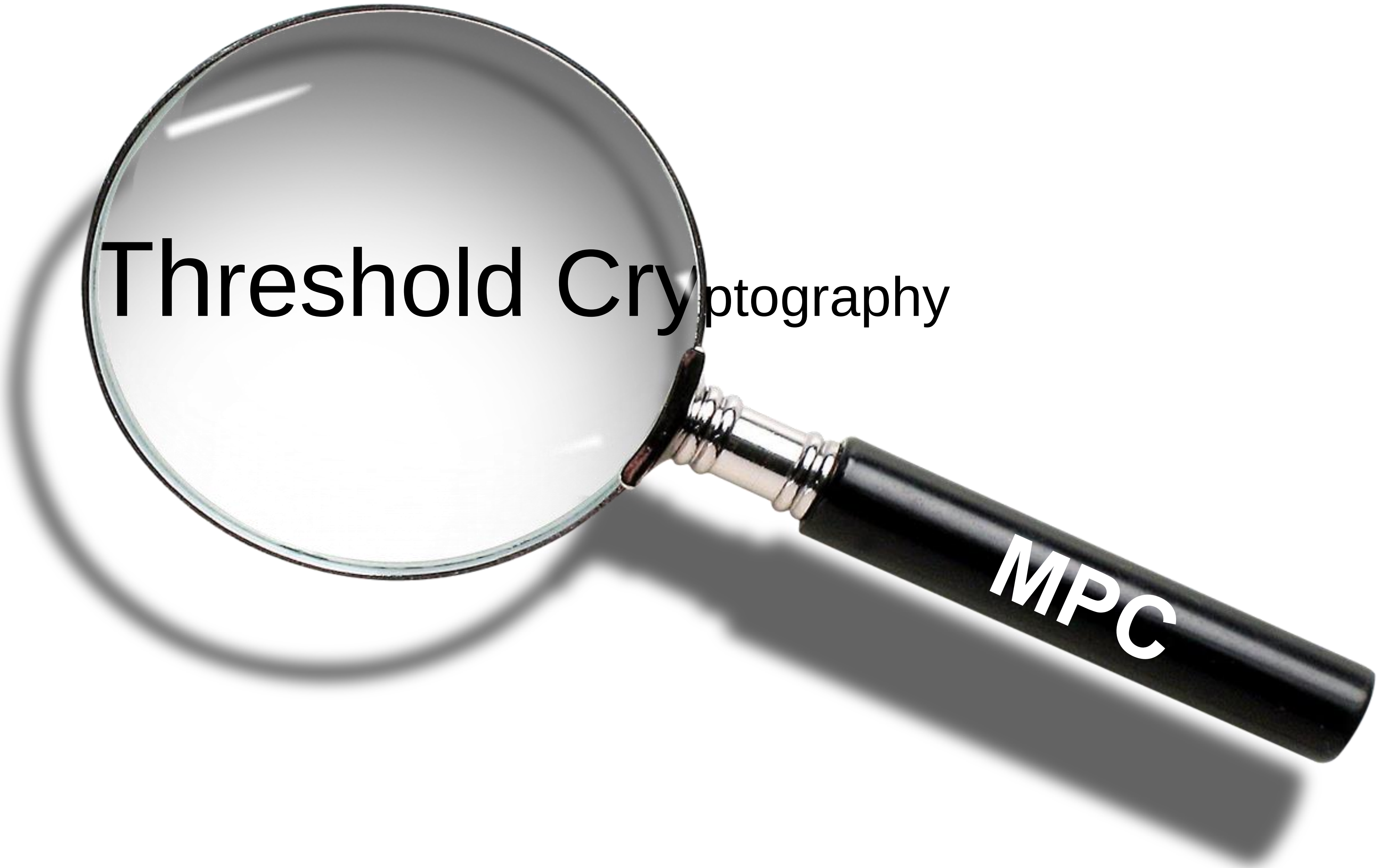
Data Warehouse

Trust in Centralized Entity

Secure Multiparty Computation (MPC) to the Rescue?

- MPC and Threshold Cryptography are clearly related, but often not mentioned in the same breath
- MPC usually deals with players who already have inputs and want to compute a joint (arbitrary) function
- Threshold cryptography wants to distribute an existing functionality – a specific goal for a specific problem

In This Talk



STEALTH SOFTWARE TECHNOLOGIES

Modeling

Implementation

Modeling

Implementation

Modeling

UC Non-Interactive, Proa
with Identifi

Ran Canetti[†‡] Rosario Gennaro[§] Ste
Udi P

October 2

Abstr

Building on the Gennaro & Goldfeder and Lindell & No
protocols, for any number of signatories and any thre
art:

- Only the last round of our protocols requires kr
  take place in a preprocessing stage, lending to a
- Our protocols withstand adaptive corruption of
  refresh mechanism and offer full proactive securi
- Our protocols realize an ideal threshold signature
  random oracle model, assuming Strong RSA, DD
  a somewhat enhanced variant of existential unfo
- Both protocols achieve accountability by identifying corrupted signatories in case of failure to generate
  a valid signature.

Simple Three-Round Multiparty Schnorr Sig with Full Simulatability

Yehuda Lindell

Coinbase
yehuda.lindell@gmail.com

Abstract. In a multiparty signing protocol, also known as a threshold
signature scheme, the private signing key is shared amongst a set of parties
and only a quorum of those parties can generate a signature. Research
on multiparty signing has been growing in popularity recently due to its
application to cryptocurrencies. Most work has focused on reducing the
number of rounds to two, and as a result: (a) are not fully simulatable
in the sense of MPC real/ideal security definitions, and/or (b) are not
secure under concurrent composition, and/or (c) utilize non-standard
assumptions of different types in their proofs of security. In this paper, we
describe a simple three-round multiparty protocol for Schnorr signatures
and prove its security. The protocol is fully simulatable, secure under
concurrent composition, and proven secure in the standard model or
random-oracle model (depending on the instantiations of the commitment
and zero-knowledge primitives). The protocol realizes an ideal Schnorr
signing functionality with perfect security in the ideal commitment and
zero-knowledge hybrid model (and thus the only assumptions needed are
for realizing these functionalities). We also show how to achieve proactive
security and identifiable abort.

Threshold ECDSA from ECDSA Assumptions: The Multiparty Case

Jack Doerner
j@ckdoerner.net
Northeastern University

Yashvanth Kondi
ykondi@ccs.neu.edu
Northeastern University

Eysa Lee
eysa@ccs.neu.edu
Northeastern University

abhi shelat
abhi@neu.edu
Northeastern University

May 22, 2020

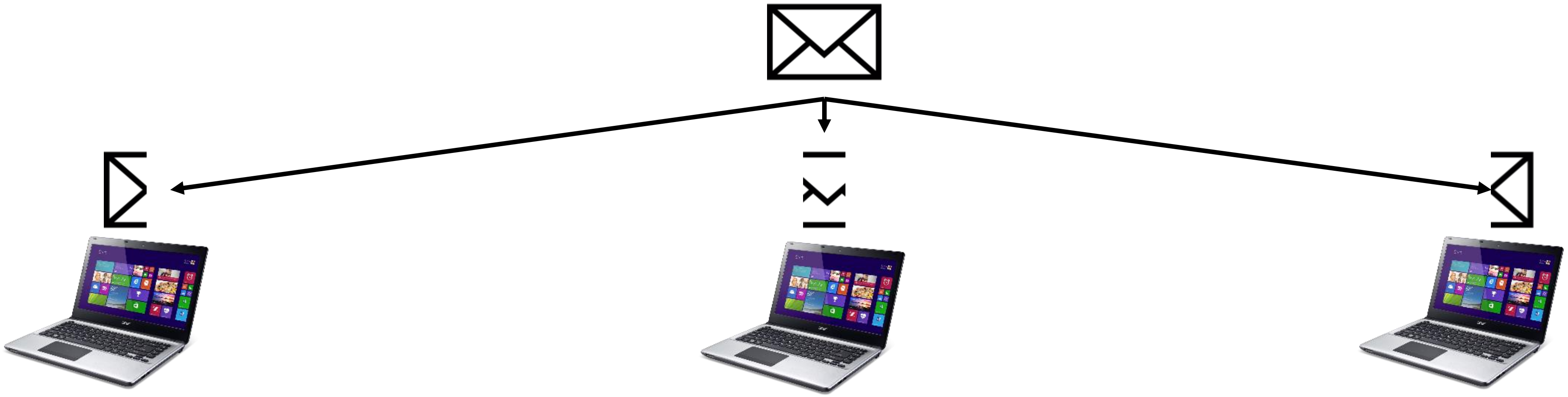
Abstract

Cryptocurrency applications have spurred a resurgence of interest in the computation of ECDSA signatures using threshold protocols—that is, protocols in which the signing key is secret-shared among n parties, of which any subset of size t must interact in order to compute a signature. Among the resulting works to date, that of Doerner et al. [DKLs18] requires the most natural assumptions while also achieving the best practical signing speed. It is, however, limited to the setting in which the threshold is two. We propose an extension of their scheme to *arbitrary* thresholds, and prove it secure against a malicious adversary corrupting up to one party less than the threshold under only the Computational Diffie-Hellman Assumption in the Global Random Oracle model, an assumption strictly weaker than those under which ECDSA is proven.

STEALTH
SOFTWARE TECHNOLOGIES

Modeling

- Modeling attributes of Threshold Cryptography can help identify and fine-tune the tradeoffs
- Obvious Attribute: Threshold (t-out-of-n)



STEALTH
SOFTWARE TECHNOLOGIES

Modeling Attributes

- Just looking at threshold ECDSA, a survey paper [AHS20]:

Majority	Corruption	Assumptions
Honest Dishonest	Static Adaptive	DDH Enhanced ECDSA etc.

Building Blocks	Rounds	Features
VSS Commitments ZK	Setup Signing On/Offline	Identifiable Abort UC etc.

Modeling Attributes

- These are the same kinds of attributes listed in MPC survey literature!
- As we standardize threshold cryptography, take a page from the categorization of MPC schemes
- Cryptographers generally do a good job writing down the model, can standardization help w/ communicating them?

Modeling

Implementation

Implementation

- Use an existing library

OR

- Implement paper from scratch, possibly releasing a library

- Decision largely influenced by ease of integration and utility to application

Implementation

- Non-distributed versions of these crypto primitives are already difficult to implement right
  - E.g. plain signature schemes have many flavors and optimizations, not to mention many buggy implementations
- Standardizing threshold cryptography should come with a lot more suggestions on getting it right
  - Interactive schemes are much more complex than primitives
- Viewing it as MPC can help modularize the problem

Implementation

- One story: working on asynchronous MPC has helped in other areas such as Private Set Intersection (PSI)
  - Boost.Asio/libevent/Netty
- In one particular instance, work with Virginia Longitudinal Data System on a (honest-but-curious) PSI system, we applied our async knowledge:
  - 1Mx1M PSI in 5 seconds, 100Mx100M in 20 minutes
  - Despite high latency between the AUS, UK, US regions
- Hypothesis: large overlap in skillset between threshold cryptography and MPC engineering

“Generality”

Threshold Schemes

- Custom computation
- Custom messages
- Faster, fewer rounds

MPC Frameworks

- Off-the-shelf algo (maybe tweaked for MPC-friendliness)
- Run MPC

Can we find more common ground?

MPC Frameworks

- This talk is sandwiched by two threshold-from-MPC talks
- Demonstrates thresholding of many existing primitives
  - Easily extends to new ones as well
- But...
  - Might be tricky to integrate or deploy
  - Still many difficult modeling questions and choices
- Work on standardizing components, as well as good guiding principles a la NIST Privacy Framework

Additional Remarks

Standardizing MPC May Help Related Areas

- Zero-Knowledge Proofs

- Private Set Intersection
  - Apple (CSAM), Signal (Contact Discovery)

- Secure Analytics
  - Prio, Google Ads, Amazon Cleanrooms

Other Considerations

- Access structures beyond t-out-of-n threshold
- Formal Verification
- etc.

Info

Dr. Steve Lu, CEO
steve@stealthsoftwareinc.com

Dr. Rafail Ostrovsky, Co-Founder
rafail@stealthsoftwareinc.com

https://stealthsoftwareinc.com

Key Takeaways

MPC offers plenty of modeling concepts that apply to threshold cryptography

Engineering can be made more modular with MPC frameworks, can we make them easier to integrate and deploy?

Standardization effort for threshold cryptography can take advantage of the two-way street it has with MPC