Secure Publish Subscribe Compute System

Anonymous Author(s)

ABSTRACT

Your abstract should go here. You will also need to upload a plain-text abstract into the web submission form.

CCS CONCEPTS

• **Security and privacy** → Use https://dl.acm.org/ccs.cfm to generate actual concepts section for your paper;

KEYWORDS

template, formatting, pickling

1 INTRODUCTION

2 RELATED WORK

3 PROTOCOL

* We minimize the communication with `Garbler`, using clever tricks, e.g., we don't need to send labels to `Garbler` for circuit garbling instead they generate it independently using shared seed.

* publishers and subscribers only talk to the broker. The communication between publishers and subscriber is done through `Broker` using end-to-end encryption. This ensures that

* Does the subscriber seed needs to be computation specific?

* We use ratcheting for forward security if seed is compromised.

* We assume PKI.

Intuition. **Initialization.**

- Each new publisher sends to `Broker` a policy specifying allowed computations on its data and generates a truly random seed s_P and send it to `Garbler`.
- All publishers and subscribers establish an authenticated encrypted channel with `Broker` and through `Broker` with `Garbler`.

Subscription.

- To subscribe computation c, subscriber sends a subscription request containing c to `Broker` and requests an output masking seed from `Garbler`.
- `Garbler` sends a truly random seed s'_c for computation c; generating a new seed if this is the first subscription for computation c.

Publication.

- To publish kth value, publisher generates two pseudorandom wire labels, w_0 and w_1, using seed s_i, for each bit of the value. w_0 is ath and w_1 is $(a+1)$th numbers in pseudorandom sequence generated using seed s_i; $2kL \leq a < 2(k+1)L$, L being the bit-length of a value.

Computation.

- After `Broker` has wire labels for all publishers' inputs required for computation c, it requests `Garbler` to garble circuit for c.
- `Garbler` independently generates input wire labels using seed s_P from each publisher contributing input and an output mask m using seed s'_c for each bit of output.
- `Garbler` generates garbled circuit for $M \circ C(.)$, composition of masking function M (XOR) and computation C, and sends it to `Broker`.
- `Broker` evaluates the garbled circuit using wire labels sent by publishers, obtains masked output $o \oplus m$, and send $o \oplus m$ to all subscribers of computation c.
- Subscribers generate the mask m using the seed s'_c and unmask the output o.

Forward Security.

Garbled Circuit XOR Compatibility. * What if one publisher doesn't send wire labels. After timeout the broker can inform the `Garbler` and it will use zero for such values in the circuit.

4 SYSTEM

5 EVALUATION

6 CONCLUSION

REFERENCES