COMP 3704 Computer Security

Christian Grothoff
christian@grothoff.org

http://grothoff.org/christian/

Secure Multiparty Computations

The general **Secure Multiparty Computation** problem is when multiple parties (say, Alice and Bob) each have private data (respectively, a and b) and seek to compute some function $f(a, b)$ without revealing to each other anything unintended.

Secure Multiparty Assumptions

We assume that all parties are semi-honest:

- They generally follow the protocol properly,
- with the exception that they keep intermediate results,
- trying to derive other partiess' private inputs.

Example: Average

1. Alice adds a secret random number to her salary and sends the sum to Bob.
2. Bob adds his salary and sends the result to Carol.
3. Carol adds her salary and sends the result to Dave.
4. Dave adds his salary and sends the result to Alice.
5. Alice subtracts the secret random number, divides and broadcasts the result to everybody.

Example: Vector Product (1/2)

1. Alice has a vector $X = (x_1, \ldots, x_n)$ and Bob has vector $Y = (y_1, \ldots, y_n)$, they agree on numbers p and m such that Bob guessing with probability $1 : p^m$ is acceptably low.
2. Alice generates m random vectors V_i such that $X = \sum_{i=1}^{m} V_i$.
3. Bob generates m random numbers r_i such that

$$v = \sum_{i=1}^{m} r_i. \tag{1}$$

Example: Vector Product (2/2)

4. For each $j = 1, \ldots, m$, Alice and Bob do:
   (a) Alice generates a secret random number k, $1 \leq k \leq p$.
   (b) Alice sends $(H_1, \ldots, H_p)$ to Bob where $H_k = V_j$ and the other H_i's are random vectors.
   (c) Bob computes $Z_{j,i} = H_i \cdot Y + r_j$ for $i = 1, \ldots, p$.
   (d) Using oblivious transfer, Alice obtains $Z_j = Z_{j,k} = V_j \cdot Y + r_j$ while Bob learns nothing about k.

5. Alice computes

$$u = \sum_{j=1}^{m} Z_j = X \cdot Y + v. \tag{2}$$

Theorem

- The general secure multi-party computation problem is solvable (using circuit evaluation protocols).
- The communication complexity of the resulting protocols depends on the size of the circuit.
- Using the solutions derived from this general approach to solve specific problems can be impractical.

Anonymity: Dining Cryptographers

"Three cryptographers are sitting down to dinner. The waiter informs them that the bill will be paid anonymously. One of the cryptographers maybe paying for dinner, or it might be the NSA. The three cryptographers respect each other's right to make an anonymous payment, but they wonder if the NSA is paying." – David Chaum

Anonymity: Definition

1. Attacker computes a **probability distribution** describing the likelyhood of each participant to be the responsible party.

2. Anonymity is the stronger, the larger the anonymity set and the more evenly distributed the subjects within that set are.

Formal Definition

Let $\mathcal{U}$ be the attacker's probability distribution describing the probability that user $u \in \Psi$ is responsible. Define the effective size S of the anonymity distribution $\mathcal{U}$ to be:

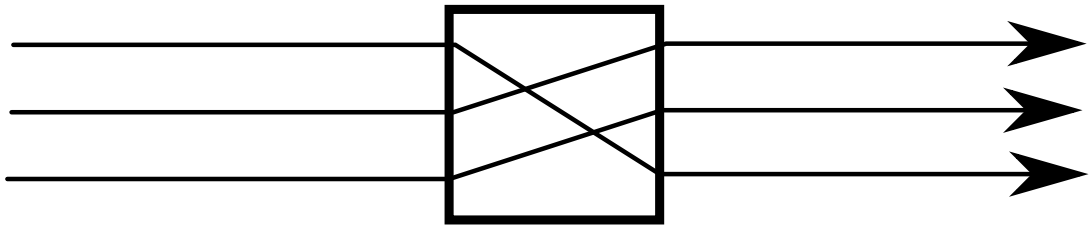
$$S = -\sum_{u \in \Psi} p_u \log_2 p_u \tag{3}$$

where $p_u = \mathcal{U}(u)$.

This is the expected number of bits of additional information that the attacker needs to definitely identify the user!

Mixing

David Chaum's mix (1981) and cascades of mixes are the traditional basis for destroying linkability:



Mixing

David Chaum's mix (1981) and cascades of mixes are the traditional basis for destroying linkability:

Secure Voting Requirements

- Only authorized voters can vote.
- No one can vote more than once.
- No one can determine for whom anyone else voted.
- No one can change anyone else's vote.
- No one can duplicate anyone else's vote.
- Every voter can make sure that his vote has been counted.

Simplistic Voting Protocol

1. Each voter signs his vote with his private key.
2. Each voter encrypts his signed vote with Trent's public key.
3. Each voter sends his encrypted vote to Trent.
4. Trent decrypts the votes, checks signatures and tabulates the votes.

Voting with Blind Signatures

1. Each voter generates n "yes" and n "no" voting messages, each with a unique random identification number.
2. Each voter gets one "yes" and one "no" ballot signed by Trent using blind signatures; Trent checks that the voter is eligible.
3. Each voter anonymously and confidentially sends one of his votes to Trent.
4. Trent decrypts, checks the signatures, checks for duplicate identification numbers, tabulates and publishes the results (including identification numbers).

Voting with Bounded Participation

1. Trent publishes a list of all legitimate voters.
2. Each voter tells Trent whether he intends to vote.
3. Trent publishes a list of voters participating in the election.
4. Voting with Blind Signatures takes place.
5. Voters verify that there were not more signatures than participants.

Digital Cash

1. Alice prepares n anonymous money orders for \$1000 each, with different random serial numbers.

2. Using (half-)blind signatures, the bank signs one of the certificates and deducts \$1000 from Alice's account.

3. Alice gives the signed certificate to a merchant in exchange for goods.

4. The merchant gives the certificate to the bank, which checks the signature, verifies that the serial number has not been used yet, and credits the merchant.

Digital Cash with Detection of Cheaters

1. Alice prepares n anonymous money orders for a given amount with a unique random serial number and m pairs of identity bits strings $I_1, \ldots, I_m$ generated using secret splitting. She commits to each pair using a bit-commitment protocol (the result becomes part of the money order).

2. Alice blinds all n money orders, the bank verifies $n-1$ of the money orders, checking the amount and identity strings. The bank signs the remaining order and deducts the amount from Alice's account.

Digital Cash with Detection of Cheaters

3. Alice gives the signed money order to a merchant; the merchant verifies the bank signature and asks Alice for a randomly selected half of each of the m identity bit strings.

4. The merchant verifies that the identity bit strings reveiled by Alice match her earlier bit-commitment and gives her the goods.

Digital Cash with Detection of Cheaters

5. The bank verifies the signature and checks the unique serial number. If the number is unique, it stores the identity bit strings and the serial number in a database.

6. If the serial number is in the database and Alice cheated, the bank has a $1 : 2^m$ chance of determining Alice's identity. If the merchant tries to cache the certificate twice, the identity bit strings will match the previous transaction.

Questions

?

Problem

Why is it unlikely that Digital Cash will ever be widely adopted (even if the protocol also provided transferability and divisibility)?