Comments on cryptographic protocols

Ian F. Blake

25 September, 2007

UBC

Introduction to secure multiparty computation

Most cryptographic protocols can be thought in terms of the following (very abstract) computational model:

- n players each have a secret piece of information
- They wish to enter into a computation of some function in which:
  - the function depends on each party's secret information
  - all parties learn the result of the computation - and are convinced of its validity
  - no one learns any other information
  - especially of each others values
- Most such functions can be realized by a Boolean function model
- Many surprising results which have **potential application in practice**
- All depend on results from computational number theory - not of interest h

Computational number theory

The number of ”useful” computational problems is very limited:

- Solving the equation $x^2 \equiv a \pmod{n}$
  equivalent to factoring the integer n
- Factoring the integer $n = p \cdot q$, $p,\ q$ primes
- Given g in some multiplicative group (integers mod p)
  and $g^x \pmod{p}$ find x
  - ▷ (the discrete logarithm problem - in other structures as well)
  - ▷ especially elliptic curves
- Not interested in these problems here
- How to use these to implement useful protocols?

Comparing two bit strings:

- Alice and Bob each have a secret bit string of the same length
- They wish to know if the bit strings are identical
- If they are not identical they learn nothing about the other
- A low tech solution - passwords, airline reservations eetc
- A high tech solution - hash functions,
  as a crypto primitive

$h: \{0,1\}^* \longrightarrow \{0,1\}^n$ preimage resistant, collision resistant etc.

Proving knowledge:

Proving knowledge:

- I have a piece of information (eg. proof of a fact, etc.)
- I want to demonstrate to you that I do in fact know what I state without divulging the proof of it
- Zero knowledge interactive proofs (ZKIP)
  - k repetitions
- e.g. Where's Waldo
- A low tech solution - cutting a copy of the picture
- Another low tech solution
- High tech solution
  - e.g. passport system based on modular square roots

Coin flipping over telephone - using only square roots

- Alice chooses $p,\ q \equiv 3 \pmod 4$ - wants square roots
- Sends $n = pq$ to Bob
- Bob chooses $x \in_R \mathbb{Z}_n$ sends $y \equiv x^2 \pmod n$ to Alice
- Alice computes square roots of $y \pmod n$, $\quad \pm x_1, \ \pm x_2$
- Alice chooses one of the four, say r, and sends it to Bob
- If $r = \pm x$ Bob loses - else Bob can factor n

The millionaires (GT) protocol

- Alice is worth X million dollars and Bob Y - who is wealthier?
- They only want the one bit of information to be known
- Specifically they don't want any information about actual values known
- A computationally inefficient algorithm to do this is known
  - based on computational number theory - homomorphic encryption

Electronic auctions and the GT protocol

- n people submit a secret bid for an item
- They want the highest bid to win the item
- They want no one to know their own bid - or any other bid
- They want to have confidence in the outcome
- Can be done without a central server - that
  with only the players exchanging information
- Repetitive use of the GT protocol
- Many variations of auctions can be done

Electronic voting

- n people enter a vote 0 or 1
- Receipt free - don't want the voter to have anything that can prove to a third party how they voted
- The voter has to be able to check at some later time their vote was counted correctly
- Such systems exist (Cryptomathic) but not suitable for large scale voting
- Secure electronic voting very difficult to implement - theoretically okay but requires sophistication on the part of the voter and large, vulnerable software systems

Electronic cash

- A client converts actual money to electronic cash (bit strings)
- Client gives a merchant the bit string representing payment
  Merchant deposits bit string to their bank who
  sends it to the clients bank for payment
- How to prevent the client (or merchant) to ”spend” the ecash again?
- If client spends it again, their identity revealed
  (by solving two equations)

Contract signing

- Two parties wish to sign a contract electronically
- How to do this so neither party can ”cheat” e.g. not send the last bit?
- The notion of ”oblivious transfer” was introduced

RSA modulus generation

- n people wish to generate a distributed RSA system
  ▷ Need a product of two primes (unknown factorization)
  ▷ public encryption exponent e
  ▷ and a secret decryption exponent d (per individual)
- They want to generate $n = pq$ (product of two primes
  no one knows the actual primes (!!)
  everyone knows the encryption exponent
  each gets a portion of the decryption exponent d
  need at least k portions to decrypt - secret sharing
- Very complicated - uses distributed statistical tests

Final comments

- Many interesting (surprising) protocols
- Most are computationally very intensive and very inefficient
- The challenge is to make them user friendly and effective
- An interesting area - private information retrieval
  The needs of the large amounts of data
  Recovering information anonymously and securely
  ▷ stored distributively/geographically
  ▷ stored encrypted
  ▷ to be retrieved privately/anonymously (PIR)
  ▷ to be retrieved error free

⇒ much to be done