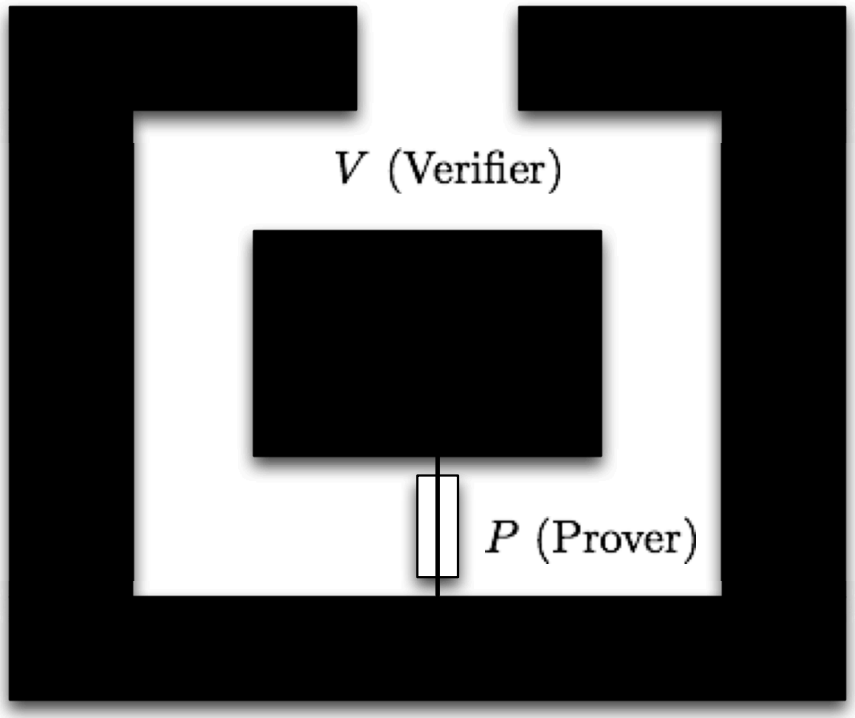
Cryptographic Protocols



Cryptographic Protocols

- A cryptographic protocol involves 2 or more parties, and performs a particular function, and may require several rounds of cryptographic computation and communication between the parties
- Its goals are usually beyond the classical goals of confidentiality, integrity or authentication
- The protocol may have several assumptions, such as the honesty of the participants (honest but curious)
- As the assumptions get weaker (allowing parties to cheat or to be malicious) the protocols get more complicated, and perhaps more useful in practice

Cryptographic Protocols

- Commitment protocols allow one to commit to a chosen value (or chosen statement) while keeping it hidden to others, with the ability to reveal the committed value later
- Zero-knowledge protocols are used to prove that a person holds an attribute (password) without revealing any information the attribute
- Secure multiparty computations allow several parties to compute a function with several inputs, so that when the protocol is complete the participants know only their own input and the final computed value
- Secret sharing protocols distribute a secret amongst a group of participants, each of whom is allocated a share of the secret, such that the secret can be reconstructed only when a sufficient number of shares are combined together

Coin Flipping over the Telephone

- The roomies Alice and Bob want to flip a coin to make a decision who will get the couch after Alice is accepted to MIT for graduate studies, and Bob will not leave UCSB (for obvious reasons)
- However, Alice is already in Cambridge, and they would like flip the coin that no party can cheat (obviously, Skype is not a good option since the video feed can be manipulated)
- The idea behind the coin flipping is that Alice will make a choice between two items Bob is offering, without knowing which one would help Bob
- Assumptions: Both parties want to win and both parties participate in the steps of the protocol

Coin Flipping over the Telephone

- Alice generates random primes p and q and computes $n = p \cdot q$
- Alice sends n to Bob
- Bob generates a random $u < n$, and computes $y = u^2 \pmod{n}$
- Bob sends y to Alice
- Alice, knowing the factors of n, solves the discrete square root problem $y = x^2 \pmod{n}$, and finds the 4 roots $x_1, -x_1, x_2, -x_2$
- One of the nontrivial square roots x_1 and x_2 is equal to u, however, Alice does not know which one, and she flips a coin (makes a choice) and sends x_1 or x_2 to Bob
- Bob receives x_i from Alice
  1. If $x_i = u$, Bob learns nothing and cannot factor n, and Bob loses
  2. If $x_i \neq u$, Bob learns a second square root of y, and thus, Bob can factor n, and therefore, Alice loses

Coin Flipping over the Telephone

- If we know two nontrivial square roots of n, we can factor n:

$$a^2 - b^2 = (a - b) \cdot (a + b) = 0 \pmod{n}$$

  implies $\gcd(n, a - b)$ or $\gcd(n, a + b)$ is a factor of n
- Alice: $n = 101 \cdot 103 = 10403 \rightarrow$ Bob
- Bob: $u = 250$, $y = u^2 = 250^2 = 82 \pmod{10403}$, and $82 \rightarrow$ Alice
- Alice: $\sqrt{82} \pmod{10403}$, and finds $x_1 = 250$, $x_2 = 4694$ (and also: $-x_1 = -250 = 10153$ and $-x_2 = -4694 = 5709$)
  1. Alice chooses 250 and $\rightarrow$ Bob
     Bob learns nothing new, and cannot factor n: Bob loses
  2. Alice chooses 4694 and $\rightarrow$ Bob; Bob peforms

$$\begin{aligned} \gcd(10403, 250 - 4694) &= \gcd(10430, 4444) = 101 \\ \gcd(10403, 250 + 4694) &= \gcd(10430, 4944) = 103 \end{aligned}$$

  Bob factors n, and $101, 103 \rightarrow$ Alice; Bob wins

Secret Sharing

- Secret sharing methods are also called "threshold cryptography"
- Threshold cryptography deals with the problem of sharing a secret among a group of users so that only when a (pre-selected) number of them come together can the secret be reconstructed
- Well-known secret sharing schemes in the literature include:
  - Shamir's method which is based on polynomial interpolation,
  - Blakley's method which is based on hyperplane geometry
  - Mignotte and Asmuth-Bloom methods which are based on the CRT
- More formally: a (t, n)-secret sharing scheme is used to distribute a secret S among n people such that any coalition of size t or more can construct d but smaller coalitions cannot

Mignotte Secret Sharing

- Setup a CRT moduli set m_i for $i = 1, 2, \ldots, n$ such that $\gcd(m_i, m_j) = 1$ for $i \neq j$ and $1 \leq i, j \leq n$
- Assume that $m_1 < m_2 < \cdots < m_n$
- The moduli set should have the property that the product of the smallest t moduli is greater than the product of the largest $t - 1$
- The secret S is selected to be in this range
- Example: $m_1 = 11$, $m_2 = 13$, $m_3 = 17$, and $m_4 = 19$
- Let $t = 3$
- The product of the smallest 3 of them: $11 \cdot 13 \cdot 17 = 2431$
- The product of the largest 2 of them: $17 \cdot 19 = 323$
- Therefore, the condition is satisfied, and the secret S will be selected in the range $323 < S < 2431$

Mignotte Secret Sharing

- The secret S is distributed among n parties by giving each party s_i such that

$$s_i = S \pmod{m_i} \quad 1 \leq i \leq n$$

- If $(t-1)$ participants get together, they cannot compute the secret using the CRT since S is larger than the product of the largest $t-1$ of the moduli
- However, if t participants get together, they can compute S since S is smaller than the product of the smallest t of the moduli

Mignotte Secret Sharing

- In our $(t, n) = (3, 4)$ threshold example, we have

$$(m_1, m_2, m_3, m_4) = (11, 13, 17, 19)$$

and

$$17 \cdot 19 = 323 < S < 2431 = 11 \cdot 13 \cdot 17$$

- Choose $S = 2013 \in (323, 2431)$, and compute and distribute the parts s_i as

$$\begin{aligned} s_1 &= 2013 = 0 \pmod{11} \\ s_2 &= 2013 = 11 \pmod{13} \\ s_3 &= 2013 = 7 \pmod{17} \\ s_4 &= 2013 = 18 \pmod{19} \end{aligned}$$

Mignotte Secret Sharing

- If any three parties get together, for example, m_1, m_3, and m_4, they can compute S using

$$S = \text{CRT}(s_1, s_3, s_4; m_1, m_3, m_4) = \text{CRT}(0, 7, 18; 11, 17, 19)$$

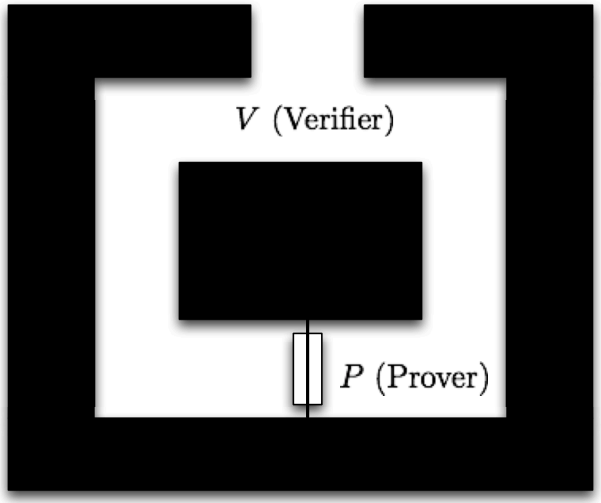
since $S < m_1 \cdot m_3 \cdot m_4 = 11 \cdot 17 \cdot 19 = 3553$

- Any two party executing the CRT, for example, m_1 and m_4 compute

$$X = \text{CRT}(s_1, s_4; m_1, m_4) = \text{CRT}(0, 18; 11, 19) = 132$$

such that $X < 11 \cdot 19 = 209$, which is $X = 132 = 2013 \pmod{209}$

Zero-Knowledge Protocol



- Prover claims the knowledge of the password that opens the door
- Prover commits by walking to Left or Right chamber
- Verifier does not know which chamber the Prover went
- Verifier flips a coin: L or R
- Verifier asks the Prover to come from L or R
- Prover succeeds by coming out from the requested side
- k rounds of success: $(1 - 2^{-k})$ probability that Prover knows

Zero-Knowledge Protocol

- Let $n = p \cdot q$, the product of two large primes
- Let y be a square mod n with $\gcd(y, n) = 1$
- Computing square roots mod n is equivalent to factoring n

- P claims to know a square root of x of y
- V will verify the claim

- P can give x to V, which is easily verified to be a square root: $x^2 = y \pmod{n}$, however, this means P gave away x
- k rounds of successful execution by P implies that P knows x with probability $1 - 2^{-k}$ or is just lucky with probability 2^{-k}

Zero-Knowledge Protocol

- P chooses a random number r_1 and computes r_2 such that $r_2 = x \cdot r_1^{-1} \pmod{n}$ which implies

$$r_1 \cdot r_2 = x \pmod{n}$$

P now computes

$$\begin{aligned} x_1 &= r_1^2 \pmod{n} \\ x_2 &= r_2^2 \pmod{n} \end{aligned}$$

and sends x_1 and x_2 to V

- V checks that $x_1 \cdot x_2 = y \pmod{n}$, and chooses either x_1 or x_2, and asks P to provide a square root of x_1 or x_2
- P gives the square root of x_1 or x_2 (which is r_1 or r_2)
- V checks to see that it is valid: $r_i^2 = x_i \pmod{n}$

Zero-Knowledge Protocol

- If P knows the square root y, he is capable of computing x_1 and x_2 such that

$$\begin{aligned} x_1 \cdot x_2 &= y \pmod{n} \\ r_1^2 \cdot r_2^2 &= (r_1 \cdot r_2)^2 \pmod{n} \\ &= x^2 \pmod{n} \end{aligned}$$

  and therefore P can immediately produce the square root of x_1 or x_2, which is r_1 or r_2, respectively

Zero-Knowledge Protocol

- If P does not know the square root x, P still can send two numbers x_1 and x_2 to V such that $x_1 \cdot x_2 = y \pmod{n}$
- However, both x_1 and x_2 cannot be square, because P does not know how to compute two square roots mod n
- Assume x_1 is a square, then, x_2 cannot be a square known to P, otherwise P would know the square root of y
- If V asks for the square root of x_1 (50% chance) then, P can provide that to V, and P wins this round (just lucky)
- If V asks for the square root of x_2 (also 50% chance) then, P cannot provide that to V, and P loses