

Lecture 11

1

RSA Signature Scheme

Use the fact that, in RSA, encryption reverses "decryption"

Let $n = pq$ where $p \neq q$ are two (large) primes
 $e \in Z_{\Phi(n)}^*$ and $e = d^{-1} \pmod{\Phi(n)}$ and $ed \equiv 1 \pmod{\Phi(n)}$
 $\Phi(n) = (p-1)(q-1)$
Secrets : p, q, d
Publics : n, e
Signing : *message* = m
Sign (x) : $y = m^d \pmod{n}$
Verification : *signature* = y
Verify (y, m) : $(m = y^e) ???$

2

RSA Signature Scheme (contd)

- The good:
 - Verification can be cheap (like RSA encryption)
 - Mechanically same as RSA decryption function
 - Security based on RSA encryption
 - Signing is harder but #verify-s > 1...
 - Deterministic
- The bad:
 - Recall that RSA is malleable: signatures can be "massaged"
 - Phony "random" signatures
 - compute $Y = \text{RSA}(e, X) = X^e \pmod n$
 - X is a signature of Y because $Y^d = X \pmod n$
- The ugly:
 - Signing requires integrity!
 - How to sign multiple blocks?
 - Deterministic - needs additional randomization!

3

El Gamal Signature Scheme

p - large prime
 b - base, generator
 x - private exponent
 y - public residue ; $y \equiv b^x \pmod p$
 $P = Z_p^*$
 $A = Z_p^* \times Z_p^*$
 publics : p, b, y
 secrets : x

Signing :

1. generate random $r \in Z_{p-1}$
2. compute $k = b^r \pmod p$
3. compute $c = (m - xk)^{-1} \pmod{p-1}$
4. signatur $e = \{k, c\}$

Verifying :

$$y^k k^c \pmod p = b^m \pmod p \quad ???$$

notice that :

$$y^k k^c = b^{xb^r} (b^r)^{(m/r - xk/r)} = b^{xb^r + m - xb^r} = b^m$$

4

El Gamal PK Cryptosystem	El Gamal Signature Scheme
<p>p – large prime b – base, primitive element, generator x – private exponent y – public residue, $y \equiv b^x \pmod{p}$ $P = Z_p^*$ $C = Z_p^* \times Z_p^*$ publics: p, b, y secrets: x</p> <p><i>Encryption:</i></p> <ol style="list-style-type: none"> 1. generate random $r \in Z_{p-1}^*$ 2. compute: $k = b^r \pmod{p}$ 3. compute: $c = my^r \pmod{p} = mb^{xr} \pmod{p}$ 4. ciphertext = $\{k, c\}$ <p><i>Decryption:</i></p> <ol style="list-style-type: none"> 1. compute $k^x \pmod{p}$ 2. compute $(k^x)^{-1} \pmod{p}$ 3. $m' = (k^x)^{-1} c = b^{-rx} mb^{xr} \pmod{p} = m$ 	<p>p – large prime b – base, generator x – private exponent y – public residue; $y \equiv b^x \pmod{p}$ $P = Z_p^*$ $A = Z_p^* \times Z_p^*$ publics: p, b, y secrets: x</p> <p><i>Signing:</i></p> <ol style="list-style-type: none"> 1. generate random $r \in Z_{p-1}^*$ 2. compute: $k = b^r \pmod{p}$ 3. compute: $c = (m - xk)r^{-1} \pmod{p-1}$ 4. signature = $\{k, c\}$ <p><i>Verifying:</i> $y^k k^c \pmod{p} = b^m \pmod{p}$???</p> <p><i>notice that:</i> $y^k k^c = b^{xb^r} (b^r)^{(m/r - xk/r)} = b^{xb^r + m - xb^r} = b^m$</p>

El Gamal Signature Scheme (contd)
<p>The good:</p> <ul style="list-style-type: none"> • Signing is cheap(er) • Designed as a signature function • Non-deterministic (randomized) <p>The bad:</p> <ul style="list-style-type: none"> • Need GOOD source of random numbers • Randomizers cannot be revealed (trace) • Randomizers cannot be reused

The Digital Signature Standard (DSS)

- Why DSS?
- RSA issues: patents, malleability, etc.
- A variant of El Gamal
- Originally for $|p|=512$ bits, now up to 1024
- Optimized for signature size (320- vs. 1024-bit)
- Signing - 1 exp, verification - 2 exps
- No attacks thus far

7

DSS (contd)

p - large prime
 b - base, generator
 x - private exponent
 y - public residue, $y \equiv b^x \pmod{p}$
 $P = Z_p^*$, $A = Z_p^* \times Z_p^*$
 publics: p, b, y secrets: x

Signing:

1. generate random $r \in Z_{p-1}^*$
2. compute: $k = b^r \pmod{p}$
3. compute: $c = (m - xk)r^{-1} \pmod{p-1}$
4. signature = $\{k, c\}$

Verifying:

$$y^k k^c \pmod{p} = b^m \pmod{p} \quad ???$$

p - 512-bit prime
 q - 160-bit prime, $(p-1)\%q = 0$
 b - base, $b^q \equiv 1 \pmod{p}$ ($b = \delta^{(p-1)/q}$)
 x - private exponent
 y - public residue; $y \equiv b^x \pmod{p}$
 $P = Z_p^*$, $A = Z_q \times Z_q$
 publics: p, q, b, y secrets: x

Signing:

1. generate random $r \in Z_{q-1}^*$
2. compute: $k = (b^r \pmod{p}) \pmod{q}$
3. compute: $c = (m + xk)r^{-1} \pmod{q}$
4. signature $e = \{k, c\}$

Verifying:

$$(b^{mc^{-1}} k^{kc^{-1}} \pmod{p}) \pmod{q} = b^k \pmod{p} \quad ???$$

notice that:

$$\begin{aligned}
 b^{mc^{-1}} k^{kc^{-1}} &= b^{mr/(m+xb^r)} (b^x)^{(b^r r)/(m+xb^r)} \\
 &= b^{(mr+xb^r r)/(m+xb^r)} = b^r
 \end{aligned}$$

8

Identification

- Public key cryptography can be also used for IDENTIFICATION
- Identification is an interactive protocol whereby one party: "prover" (who claims to be, say, Alice) convinces the other party: "verifier" (Bob) that she is indeed Alice
- Identification can be accomplished with public key digital signatures
- However, signatures reveal information...
- Also, signatures are "transferable", i.e., anyone can verify them

9

Fiat-Shamir Identification Scheme

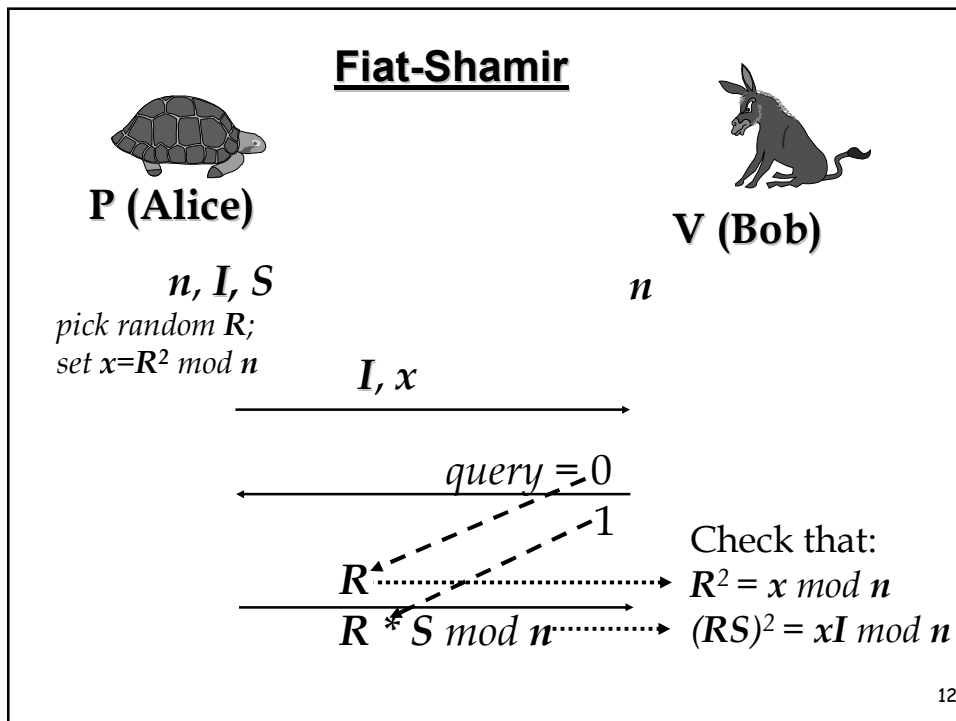
- In Fiat-Shamir, prover has an RSA modulus $n = pq$ (factorization is secret).
- Factors themselves are not used in the protocol.
- Unlike RSA, a trusted center can generate a global n , used by everyone, as long as nobody knows its factorization. Trusted center can "forget" the factorization after computing n .

10

Fiat-Shamir Identification Scheme

- Secret Key: Prover (P) chooses a random value $1 < S < n$ (to serve as the key) such that $\gcd(S,n) = 1$
- Public Key: P computes $I=S^2 \bmod n$, publishes (I,n) as his public key.
- Purpose of the protocol: P has to convince verifier (V) that he knows the secret S corresponding to the public key (I,n) ,
 - i.e., to prove that he knows a square root of $I \bmod n$, without revealing S or any portion thereof

11



Fiat-Shamir Identification Scheme

V wants to authenticate identity of P, who claims to have a public key I. Thus, V asks P to convince him that P knows the secret key S corresponding to I.

1. P chooses at random $1 < R < n$ and computes:
 $X = R^2 \pmod n$
2. P sends X to V
3. V randomly requests from P one of two things (0 or 1):
 - (a) R
 - or
 - (b) $RS \pmod n$
4. P sends requested information

13

Fiat-Shamir ZK Identification Scheme

5. V checks the correct answer:
 - a) $R^2 \stackrel{?}{=} X \pmod n$
 - or
 - b) $(R*S)^2 \stackrel{?}{=} X*I \pmod n$
6. If verification fails, V concludes that P does not know S
7. Protocol is repeated t (usually 20, 30, or log n) times, and, if each one succeeds, V concludes that P is the claimed party.

14

Fiat-Shamir Identification Scheme

CLAIM: Protocol does not reveal ANY information about S or
Protocol is **ZERO-KNOWLEDGE**

Proof: We show that no information on S is revealed:

- Clearly, when P sends X or R , he does not reveal any information on S .
- When P sends $RS \bmod n$:
 - $RS \bmod n$ is random, since R is random and $\gcd(S, n) = 1$.
 - If adversary can compute any information on S from

I, n, X and $RS \bmod n$

he can also compute the same information on S from I and n , since he can choose a random $T = R'S \bmod n$ and compute:

$$X' = T^2 I^{-1} = (R')^2 S^2 I^{-1} = (R')^2$$

15

Security

Clearly, if P knows S , then V is convinced of his identity.

If P does not know S , he can either:

1. know R , but not $RS \bmod n$. Since he is choosing R , he cannot multiply it by the unknown value S or
2. choose $RS \bmod n$, and thus can answer the second question: $RS \bmod n$. But, in this case, he cannot answer the first question R , since he needs to divide by the unknown S .

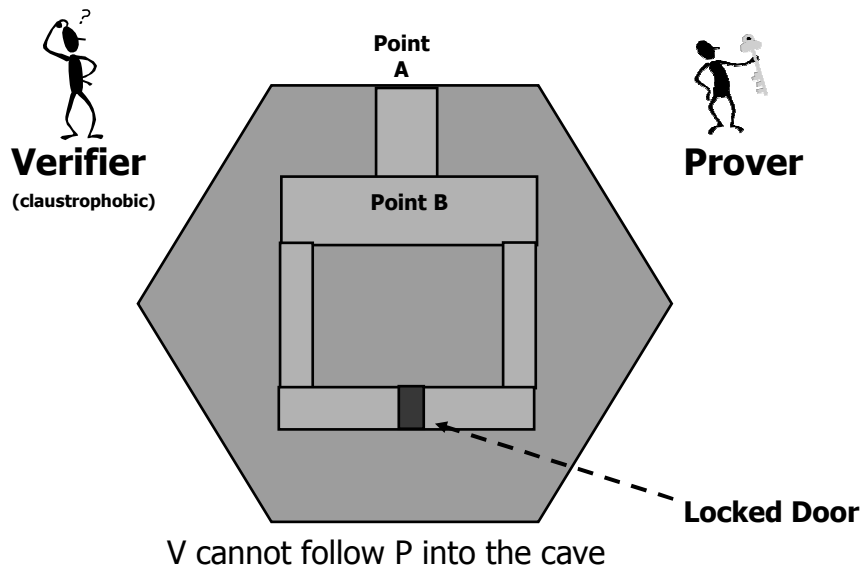
16

Security

- In any case, adversary cannot answer both questions, since otherwise he can compute S as the ratio between the two answers.
- But, we assumed that computing S is hard, equivalent to factoring n .
- Since P does not know in advance (when choosing R or $RS \bmod n$) which question that V will ask, he cannot foresee the required choice. He can succeed in guessing V 's question with probability $1/2$ for each question.
- The probability that V fails to catch P in all runs is thus: 2^{-t} (e.g., 1 in 1,000,000,000 for $t=20$)

17

How to explain ZK to your children



18

How to explain ZK to your children

The Protocol:

- 1) V checks that door is locked and comes out to point A and looks away
- 2) P goes into the maze past point B (either right or left)
- 3) V walks up to point A
- 4) V randomly picks right or left
- 5) V shouts (very loudly!) for P to come out from the picked direction

REPEAT n TIMES

