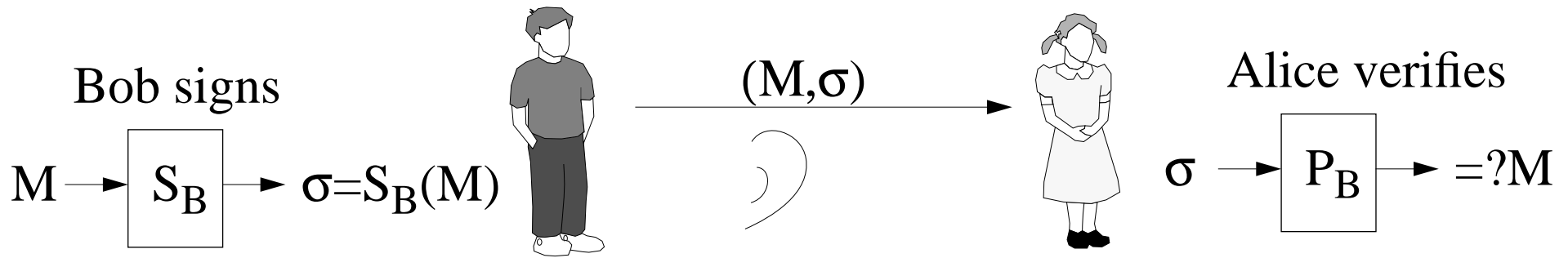


Digital Signatures using RSA

suppose you didn't care about eavesdropper, but wanted to make sure no one can forge a message to look like it came from you
(credit card or check transactions over the Internet)



The discussion for the cryptosystem works.

Also, can prove that $P(S(M))=M \pmod n$

How hard is it to compute $P_A(M)$ and $S_A(C)$?

example: $S(C)=C^{103} \pmod{143}$

do $\{[(C^2 \pmod{143})^2 \pmod{143}]^2 \pmod{143}\}^2 \pmod{143}$,etc
takes $O(\log d)$ time -- which isn't too long

How hard is it for someone who knows C and P_A to figure out S_A or $S_A(C)$?

no one knows the answer to this, but people think that it is very hard
if factoring a large number is easy,

then we could factor n into its prime factors and then figure out S_A

but if factoring is hard, it may still be easy to figure out S_A

conclusion: people have been trying to crack RSA since 1977 and haven't
been able to

Theorem: RSA works

Proof:

$$S(P(M))=S(M^e \pmod n)=(M^e \pmod n)^d \pmod n =M^{ed} \pmod n$$

e and d are multiplicative inverses modulo $(p-1)(q-1)$, i.e. $ed=1+k(p-1)(q-1)$

We use 2 theorems from number theory:

1) if p is prime then $a^{p-1} \equiv 1 \pmod p$

2) if p and q are relative prime, $M^{ed} \equiv M \pmod p$ and $M^{ed} \equiv M \pmod q$,

then $M^{ed} \equiv M \pmod{pq}$, i.e. $M^{ed} \equiv M \pmod n$

So, $M^{ed} \equiv M^{1+k(p-1)(q-1)} \pmod p$

$$M^{ed} \equiv M \left(M^{p-1} \right)^{k(q-1)} \pmod p$$

$$M^{ed} \equiv M (1)^{k(q-1)} \pmod p$$

$$M^{ed} \equiv M \pmod p$$

Similarly, $M^{ed} \equiv M \pmod q$. Thus, $M^{ed} \equiv M \pmod n$

RSA public-key cryptosystem

1) select at random two large prime numbers p and q .

p and q should each be about 200 digits long

very simple example: $p=13$ $q=11$

2) compute $n=pq$

very simple example: $n=143$

3) select a small odd integer e that is relatively prime to $(p-1)(q-1)$

very simple example: $(p-1)(q-1)=12*10=120$; $e=7$

4) compute d as the multiplicative inverse of e , modulo $(p-1)(q-1)$

recall: $x \bmod y = x - \lfloor x/y \rfloor y$, $x \equiv y \bmod z$ is $x \bmod z = y \bmod z$

very simple example: need $d*7 \bmod 120 = 1 \bmod 120$,

$d=103$ works: $103*7=721$, $721 \bmod 120 = 1$

5) publish the pair $P=(e,n)$ as your RSA public key

very simple example: $P=(7,143)$

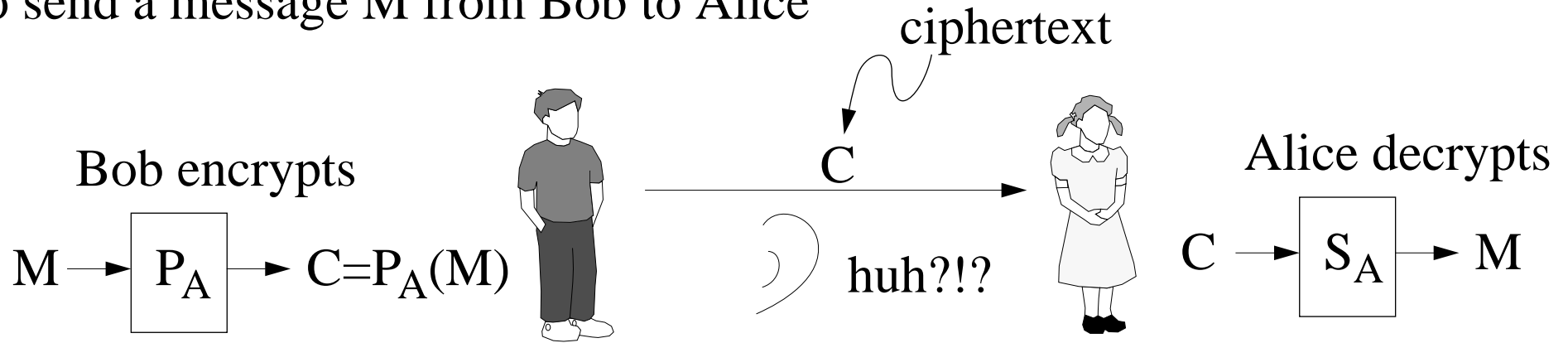
6) keep secret the pair $S=(d,n)$

very simple example: $S=(103,143)$

To encrypt, $P(M)=M^e \pmod n$

To decrypt, $S(C)=C^d \pmod n$

To send a message M from Bob to Alice



What do we need for this to work:

- I) $S_A(P_A(M)) = M$, i.e. P_A and S_A are inverse functions of each other
- II) $P_A(M)$ and $S_A(C)$ can be computed easily
- III) someone who knows C and P_A cannot figure out S_A or $S_A(C)$

Cryptosystem: a method for encrypting and decrypting

RSA cryptosystem: depends on the dramatic difference between the ease of finding large prime numbers and the difficulty of factoring the product of two large prime numbers

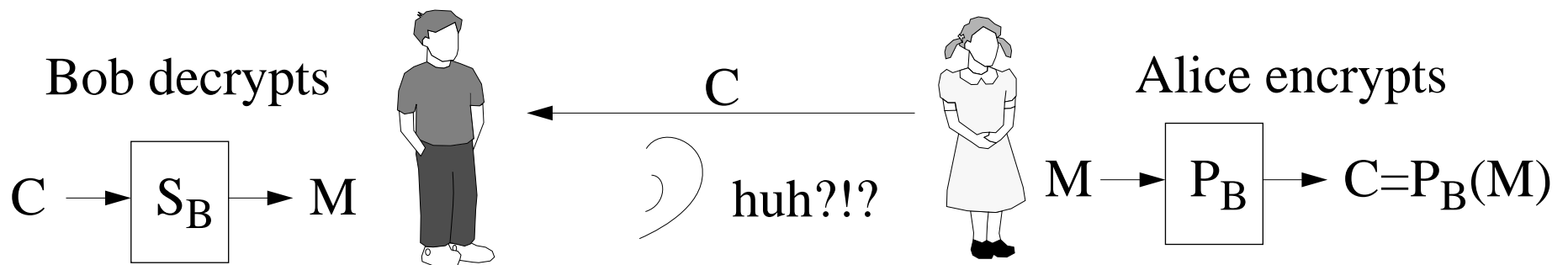
Public-key cryptosystem: each person has a public key and a secret key

Alice has P_A and S_A , her public and secret keys

Bob has P_B and S_B , his public and secret keys

Each person creates their own public and private keys (we'll say how later); they then publish their public key and keep the secret key secret.

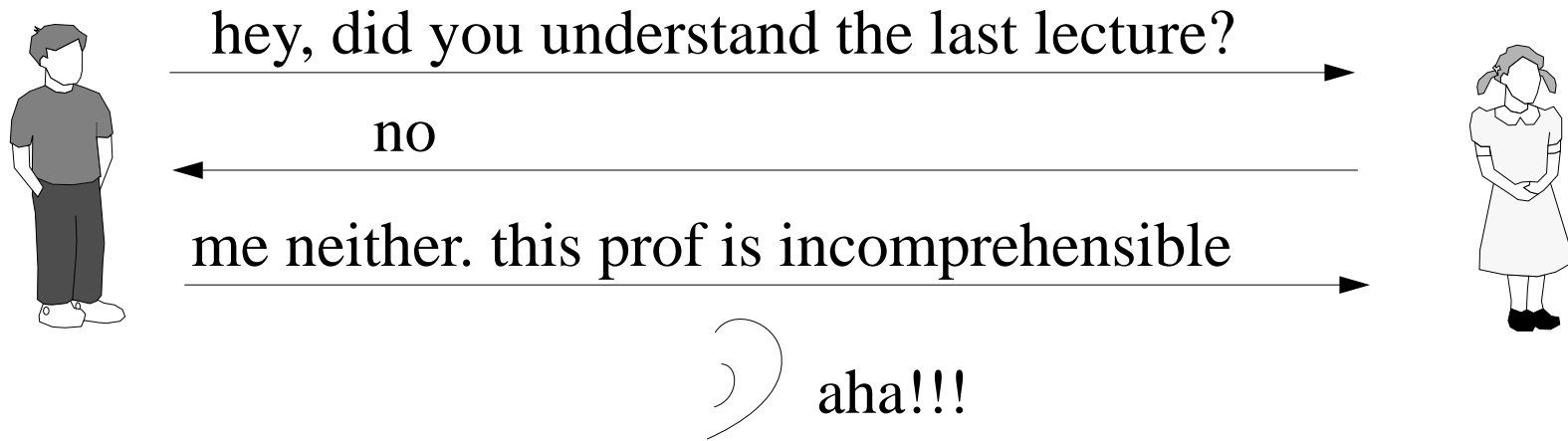
To send a message M from Alice to Bob,



TOPIC 22: Cryptography

RSA public-key cryptosystem

problem: alice and bob want to communicate privately
so that an eavesdropper who overhears their communication
cannot understand it



answer: they encrypt their messages to each other

