COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Communications Privacy III: Paradigms of Cryptography

Information Privacy with Applications

David Sidi (dsidi@email.arizona.edu)

Small mention of interesting things

- Prof. Diana Daly is here
- Power of privacy documentary (link)
- European commission of human rights: do UK surveillance laws violate human rights? (link)
- Hints for using stem in Assignment 2

Warm-up

- Explain what a one-way function is

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Continuing last time: RSA

"Cryptography is the only technique capable of providing security to messages transmitted over channels entirely out of the control of either the sender or the receiver." (POTL 35)

Rivest's riposte

- Diffie missed something: Ron Rivest's idea of chaffing and winnowing for confidentiality
- OK, still cryptography, but not encryption, so an important qualification to Diffie's comment
- Not encryption!
- Actually, he missed two things: What is another example of an approach to confidentiality that does not use encryption?

Rivest's Repost

- Chaffing and Winnowing (C&W) arose amid the same concerns about key escrow, clipper chips, etc. that POTL had in mind
- Key idea: Uses obfuscation to achieve confidentiality over an insecure channel
- A kind of compulsion resistance for cryptography development!

Chaffing and Winnowing merges integrity and confidentiality

- Sending a message has two parts
  - authenticating (adding MACs)
  - adding “chaff”
- Receiving a message requires removing the “chaff”

Chaff is a set of fake packets

- Chaff packets are not part of the real message
- The MAC of chaff doesn't check, so intended recipients can discard them

(1,Hi Larry,532105)
(1,Hi Bob,465231)
(2,Meet me at,782290)
(2,I'll call you at,793122)
(3,6PM,891231)
(3,7PM,344287)
(4,Yours-Susan,553419)
(4,Love-Alice,312265)

Senders append MACs

- Message is broken into packets by the sender
- MACs are appended to each packet (note: packet is still in the clear)
- MAC is a function of a hash of the message contents, and a shared authentication key
- a serial number can also be added

Chaffing and Winnowing

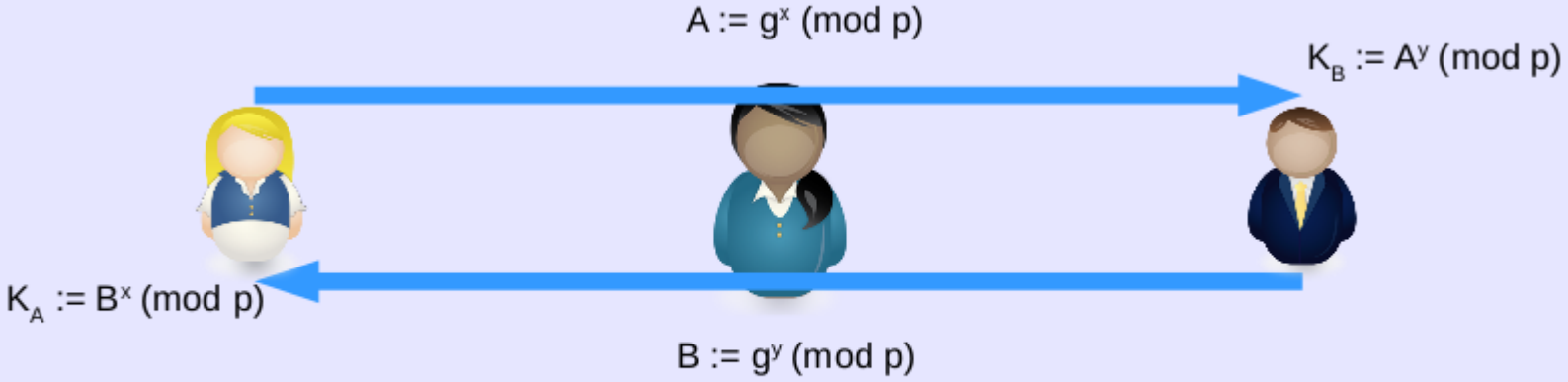
- Confidentiality of C&W depends on the MAC algorithm, on how the original message is broken into packets, and on how the chaffing is done
- MAC should be indistinguishable from a random function

Public Key Cryptography

- Key idea: Encryption key is public, decryption key is private
- “Asymmetric”
- Due to Diffie and Hellman

Public Key Cryptography

- Key idea: Encryption key is public, decryption key is private
- “Asymmetric”
- Question: Explain how DH is a “public key distribution system”



$K_A = (g^y)^x = (g^x)^y = K_B$ is the key

Historical Sidenote: Diffie-Hellman-Gill Key Agreement?

- Commonly known that Merkle had ideas of PKC as well as Diffie and Hellman
- Less well known: “Another potential one-way function, of interest in the analysis of algorithms, is exponentiation mod q, which was suggested to the authors by Prof. John Gill of Stanford University.”
  - From “New Directions”!

Public Key Cryptography

- Key idea: Encryption key is public, decryption key is private
- “asymmetric”
- Question: The public key can also be used for decryption, and the private one for encryption. When?

Big picture: Complexity results are fundamental to PKC

- *One-way functions* are operations that are “easy” in one direction, and “difficult” (better: “apparently difficult”) in the other
- Computational assumption: adversary cannot solve the “hard” problem

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

The Rivest-Shamir-Adleman (RSA) Cryptosystem

RSA is used all over the place

Key exchange/agreement and authentication

Algorithm	SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3 (Draft)	Status
RSA	Yes	Yes	Yes	Yes	Yes	No	Defined for TLS 1.2 in RFCs
DH-RSA	No	Yes	Yes	Yes	Yes	No	
DHE-RSA (forward secrecy)	No	Yes	Yes	Yes	Yes	Yes	
ECDH-RSA	No	No	Yes	Yes	Yes	No	
ECDHE-RSA (forward secrecy)	No	No	Yes	Yes	Yes	Yes	
DH-DSS	No	Yes	Yes	Yes	Yes	No	
DHE-DSS (forward secrecy)	No	Yes	Yes	Yes	Yes	No[31]	
ECDH-ECDSA	No	No	Yes	Yes	Yes	No	
ECDHE-ECDSA (forward secrecy)	No	No	Yes	Yes	Yes	Yes	
PSK	No	No	Yes	Yes	Yes		
PSK-RSA	No	No	Yes	Yes	Yes		
DHE-PSK (forward secrecy)	No	No	Yes	Yes	Yes		
ECDHE-PSK (forward secrecy)	No	No	Yes	Yes	Yes		
SRP	No	No	Yes	Yes	Yes		
SRP-DSS	No	No	Yes	Yes	Yes		
SRP-RSA	No	No	Yes	Yes	Yes		
Kerberos	No	No	Yes	Yes	Yes		

RSA is used all over the place

Quick Information

Instructor:	David Sidi
Office Location:	HARV 456
Office Hours:	Fridays at 10:30am
Telephone:	(520) 621-3565
Email:	dsidi@email.arizona.edu
OpenPGP:	4096R/0x84969123EEBA824
Home page:	https://u.arizona.edu/~dsidi/

```
$ gpg --gen-key
gpg (GnuPG) 1.4.18; Copyright (C) 2014 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 4096
...
```

RSA is used all over the place

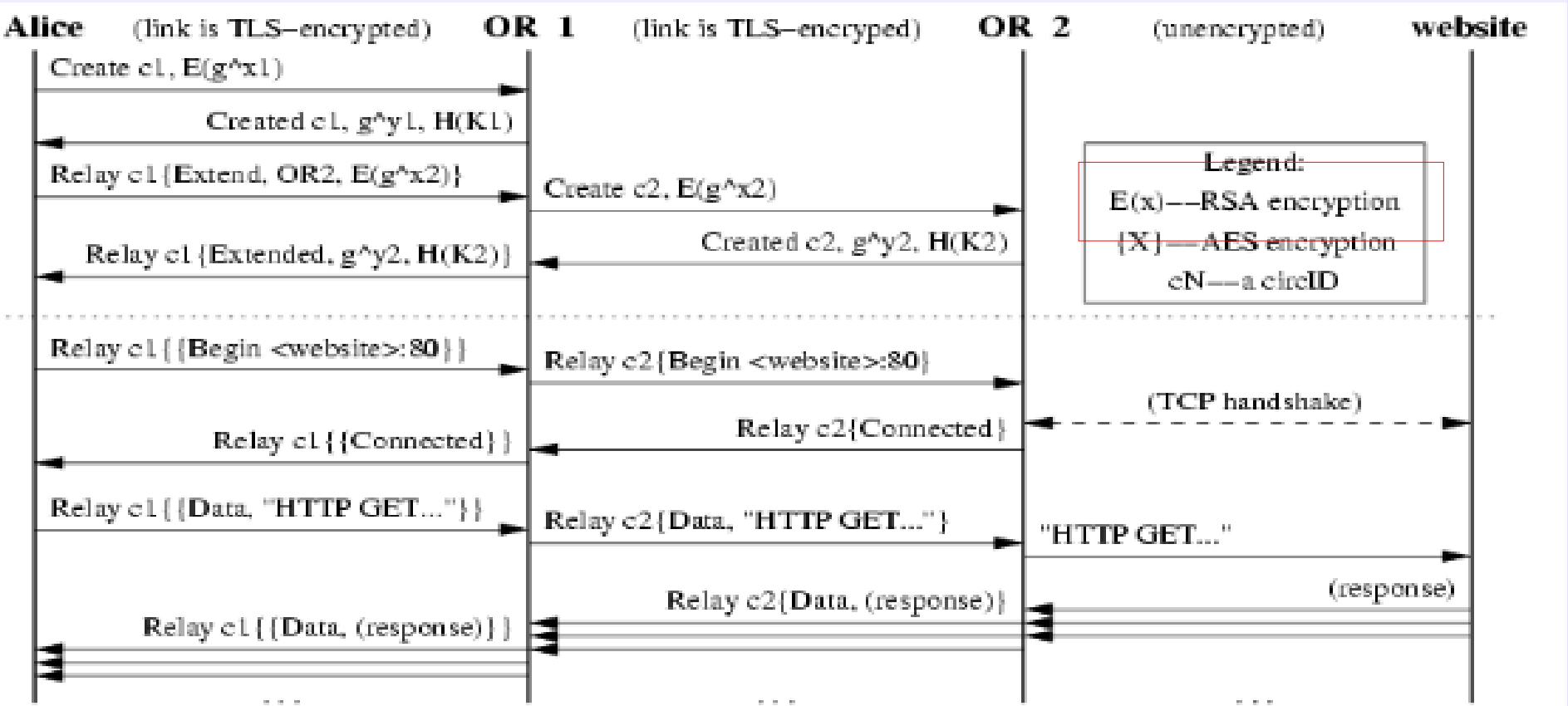


Figure 1: Alice builds a two-hop circuit and begins fetching a web page.

RSA requires a modulus that is the product of two primes

- randomly choose a large integer n = pq, called the RSA modulus, with p and q prime
- p and q of almost equal length
- There are factoring algorithms that do better with p or q of a special form, but there are only a few instances of that form. With a cryptographic pseudo-random number generator (CPRNG), the probability of getting one is negligible

Generating p and q is fraught

- Recent news: ROCA. RSA weakness found in keys generated by Infineon TPMs and smart cards)
  - “I've completed a full scan of the crt.sh DB (CT log search engine), which found 171 certs with ROCA fingerprints. The list is at https://misissued.com/batch/28/”
  - mozilla-dev-security-policy@lists.mozilla.org

RSA requires an encryption exponent

- $\varphi(n) = (p - 1)(q - 1)$ is Euler's phi (this is the order of $(\mathbb{Z}/n\mathbb{Z})^*$)
- choose an encryption exponent e such that
  - $1 <= e <= \varphi(n)$
  - e coprime with $\varphi(n)$: $\gcd(e, \varphi(n)) = 1$

RSA requires a decryption exponent

- compute a decryption exponent d
- 1 <= d <= φ(n)
- ed ≡ 1 (mod φ(n))
  - found with extended euclidean algorithm, since gcd(e, φ(n)) = 1

RSA encrypts messages encoded as integers

- message is an integer m with $0 < m < n$
  - can encode $m_1 m_2 \cdots m_k$ as such an m; a block version of RSA
- encryption of a message m is $m^e \pmod n$; decryption is $(m^e)^d \pmod n$
- we need to show that $(m^e)^d = m$

RSA relies on the difficulty of prime factorization

- choose a large integer n = pq, with p and q prime
- φ(n) is Euler's phi
- choose exponents e,d such that
  - 1 <= e,d <= φ(n)
  - e coprime with φ(n)
  - ed ≡ 1 (mod φ(n))
- message is an integer m with 0 < m < n
- encryption of m is

  $m^e \pmod n$
- decryption is

  $(m^e)^d \pmod n$

RSA is a cryptosystem

- To show that RSA is a cryptosystem, we need to show that the encryption operation is invertible

RSA is a cryptosystem

- $ed \equiv 1 \pmod{\varphi(n)}$ implies $ed = 1 + \ell\varphi(n)$
- $(m^e)^d = (m^{1+\ell\varphi(n)}) = m(m^{\ell\varphi(n)}) = m(m^{\ell(p-1)(q-1)}) = m(m^{(p-1)})^{\ell(q-1)}$
- If $p \mid m$, $(m^e)^d \equiv m \pmod{p}$ is trivial. (why?)
- Otherwise, by Fermat's little theorem, $m^{(p-1)} \equiv 1 \pmod{p}$, so $m(m^{(p-1)})^{\ell(q-1)} \equiv m \pmod{p}$
- The case is exactly similar for q, so we have $(m^e)^d \equiv m \pmod{pq}$
- $0 < m < pq$, so it is established that $(m^e)^d = m$

The algebra behind RSA

- Lots of the details you've just seen about the usual implementation of RSA are inessential
- The core idea of RSA is algebraic: exponentiation by an element coprime with the group order is an invertible automorphism; it can be done efficiently, whereas the inverse apparently cannot be
  - a different example: elliptic curves
- Klaus Lux occasionally teaches a cryptography course here; take it to learn more

RSA is a partially-homomorphic system

- What is homomorphic encryption?
- $(gh)^e = g^e h^e$, given that we're in an abelian group, so raw RSA is a multiplicatively homomorphic system

Big picture: Complexity results are fundamental to PKC

- "One-way functions:" *Computationally* noninvertible functions (apparently)
- "Computational assumption:" adversary cannot solve the hard problem

Diffie's motivation for Public Key Cryptography (PKC)

- Previously, cryptography was used to extend the security of a channel to other (higher throughput, lower latency) channels
- "The effect has been to limit the use of cryptography to communications among people who have made prior preparation for cryptographic security." (647)
- Communication is now important between parties that don't know each other, and are not within earshot of each other. Commercial examples abound

Diffie's motivation for Public Key Cryptography (PKC)

- There's a need for cryptography to ensure confidentiality of communication over insecure channels
- Key management for classical cryptographic systems doesn't scale well for the variety of commercial applications cryptography will now have
  - N users is how many keys, for traditional cryptography (e.g., OTP)?

Diffie's motivation for Public Key Cryptography (PKC)

- Key management is centralized only for the public key; private keys are kept secret by the individual key holders
  - N users, how many keys are needed?
- But private key must still be securely generated, and kept safe
  - ROCA, lest we forget. And many more
  - This will come up also with "decoy-based" approaches

Diffie's motivation for Public Key Cryptography (PKC)

- Communication is now important between parties that don't know each other, and are not within earshot of each other. Commercial examples abound.
- Cannot rely on secure channels for key exchange in advance
- Similarly, there is a need for authentication between such parties: digital signatures

Diffie's "other" idea

- Compiled programs can be made hard to reverse.
- This means it can be hard to know what function they compute; an unknown function is hard to invert!
- What about Kerchkoffs's law? (654)

Diffie's "other" idea

- "A more practical approach to finding a pair of easily computed inverse algorithms E and D; such that, D is hard to infer from E, makes use of the difficulty of analyzing programs in low level languages ...

... Anyone who has tried to determine what operation is accomplished by someone else's machine language program knows that E itself (i.e., what E does) can be hard to infer from an algorithm for E. If the program were to be made purposefully confusing through addition of unneeded variables and statements, then determining an inverse algorithm could be made very difficult. Of course, E must be complicated enough to prevent its identification from input-output pairs." (648)

An example obfuscating transformation: control-flow flattening

- fast modular exponentiation (where have we seen that?)

```
int modexp(int y,int x[],
           int w,int n) {
   int R, L;
   int k = 0;
   int s = 1;
   while (k < w) {
      if (x[k] == 1)
         R = (s*y) % n;          B6
      else
         R = s;
      s = R*R % n;
      L = R;
      k++;
   }
   return L;
}
```

```
int modexp(int y, int x[], int w, int n) {
   int R, L, k, s;
   int next=0;
   for(;;)
      switch(next) {
         case 0 : k=0; s=1; next=1; break;
         case 1 : if (k<w) next=2; else next=6; break;
         case 2 : if (x[k]==1) next=3; else next=4; break;
         case 3 : R=(s*y)%n; next=5; break;
         case 4 : R=s; next=5; break;
         case 5 : s=R*R%n; L=R; k++; next=1; break;
         case 6 : return L;
      }
}
```

Collberg and Nagra, Surreptitious Software

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Return to the Big picture: Complexity results are fundamental to PKC

- "One-way functions:" *Computationally* noninvertible functions (apparently)
- "Computational assumption:" adversary cannot solve the hard problem

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Return to the Big picture: Complexity results are fundamental to PKC

- "One-way functions:" *Computationally* noninvertible functions (apparently)
- "Computational assumption:" adversary cannot solve the hard problem
- Is this the only way? Must computational difficulty be at the heart of things?

Decoy-based cryptosystem

"there are secure encryption protocols that do not employ any one-way functions, but instead rely in their security on numerous 'decoys' of the actual encrypted message, and this 'decoy-based' cryptography presents an important alternative to the 'traditional', complexity-based, cryptography."

(Grigoriev and Shpilrain 2)

- To keep in mind: How will this compare Chaffing and Winnowing?

Decoy-based cryptosystem

"the general idea of decoy ...[is] combining private keys of Alice and Bob during transmission."

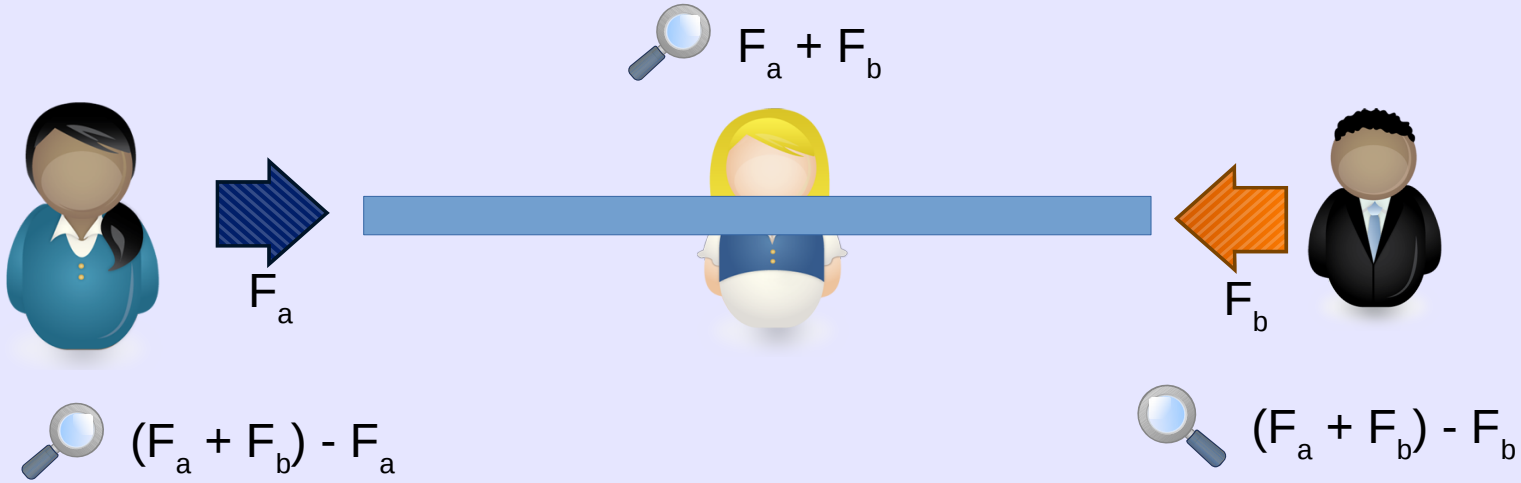
(Grigoriev and Shpilrain 2)

- To keep in mind: how does this compare to DH key agreement?

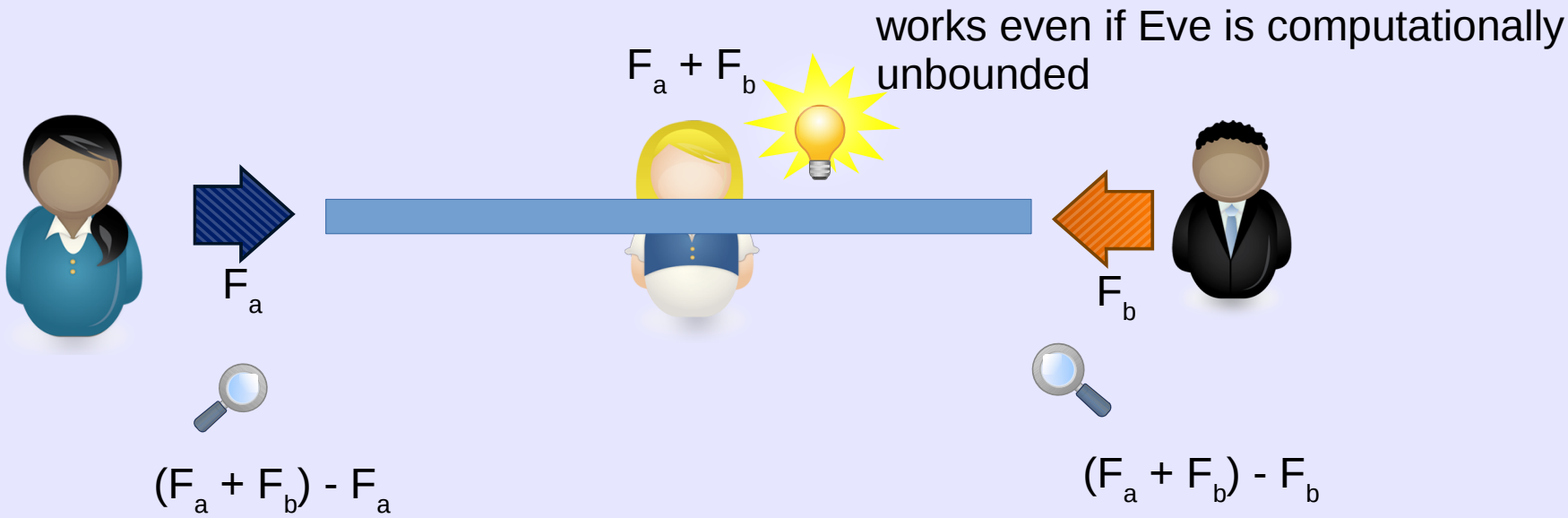
Requirements of Decoy-based Cryptography

- Requirement: a “private space,” analogous to the private computer that generates keys unobserved
- This is at the end point; it is not a reversion to requiring a secure channel
- In “public” everything is observable

A decoy implementation that resembles “Physical DH”



A decoy implementation that resembles “Physical DH”



Secure multiparty communication (SMC)

- Up to now, we have been keeping messages private from a third party, but the communicating parties trust one another
- What if you don’t trust who you’re communicating with, but you need to get something done together?
  - What if you want to compute something with inputs from you and another party, but keep your input to the computation hidden, and not rely on any trusted third party?

An SMC problem: Yao's millionaires

- Two millionaires have net worths N_1 and N_2
- They are interested in the difference in their net worths, but neither wants to share their particular net worth
- Correctness: we want to compute $d = N_1 - N_2$
- Privacy: we do not want party 2 to learn anything more about N_1 from following the protocol than they would by simply learning d

A decoy implementation that resembles “Physical SMC”

- Requirement: a “private space,” analogous to the private computer that generates keys unobserved
- This is at the end point; it is not a reversion to requiring a secure channel
- In “public” everything is observable

Simple decoy implementation