

# Louis, Lester and Pierre: Three Protocols for Location Privacy

Ge Zhong   Ian Goldberg   Urs Hengartner

David R. Cheriton School of Computer Science  
University of Waterloo

PET 2007



# Problem Statement

- ▶ Location-based services enable social-networking applications
  - ▶ E.g., (distributed) buddy tracking
- ▶ Location privacy is important
- ▶ Our focus: **Nearby-friend problem**
  - ▶ **Release location to friend iff friend is nearby**
  - ▶ Current buddy-tracking applications do not satisfy this condition

## Related Work

- ▶ Køien and Oleshchuk's location inclusion protocol based on secure 2-party computation (PET 2005)
  - ▶ Allows Alice to learn whether Bob's location is in a polygon determined by Alice
  - ▶ Problem: by providing a degenerate polygon, Alice can learn Bob's exact location
- ▶ Atallah and Du's solution based on secure multi-party computation (WADS 2005)
  - ▶ High communication cost

## Contribution

- ▶ Three protocols for learning of nearby friends—Louis, Lester and Pierre
  - ▶ Distributed
  - ▶ No third party that become aware of people's locations
  - ▶ All protocols have practical communication and computation complexity
  - ▶ Each protocol has different features

# Outline

## Overview

- Problem Statement
- Related Work
- Contribution

## Protocols

- Threat Model
- Basic Approach
- Pierre Protocol

## Analysis

- Implementation
- Performance
- Comparison

## Conclusion

# Threat Model

- ▶ Alice and Bob can learn the location of each other only if they are within distance  $r$
- ▶ No third party can learn their locations
- ▶ Alice and Bob are trusted to input their true locations
  - ▶ Our protocols can discourage them to input fake locations

# Homomorphic Encryption

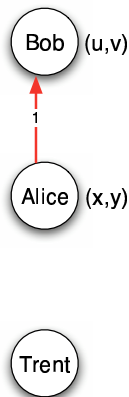
- ▶ Public-key cryptosystem with additive homomorphism
  - ▶  $\mathcal{E}(m_1) \oplus \mathcal{E}(m_2) = \mathcal{E}(m_1 + m_2)$
  - ▶  $\mathcal{E}(m * n) = \underbrace{\mathcal{E}(m) \oplus \mathcal{E}(m) \oplus \dots \oplus \mathcal{E}(m)}_{n \text{ times}}$
- ▶ Two systems used in our protocols
  - ▶ Paillier, CGS97
- ▶  $\mathcal{E}_A(\cdot)$ : encryption function with Alice's public key

## Common Technique

- ▶ Location: Alice  $(x, y)$ , Bob  $(u, v)$
- ▶ Radius  $r$  agreed by both parties
- ▶ Nearby if  $d = (x - u)^2 + (y - v)^2 - r^2 < 0$
- ▶ Have Bob compute  $\mathcal{E}_A(d) = \frac{\mathcal{E}_A(x^2+y^2) \cdot \mathcal{E}_A(u^2+v^2)}{\mathcal{E}_A(2xu) \cdot \mathcal{E}_A(2yv) \cdot \mathcal{E}_A(r^2)}$
- ▶ Need to ensure Alice (and Bob) can learn the value of  $d$  only if  $d < 0$

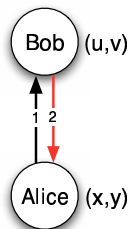
# Louis Protocol

1. Alice sends her encrypted location information to Bob
2. Bob picks a random value  $k$ , and sends  $\mathcal{E}_A(d + k)$ ,  $\mathcal{E}_T(k)$  to Alice
3. Alice sends  $d + k$ ,  $\mathcal{E}_T(k)$  to semi-trusted third party, Trent
4. Trent checks the sign of  $d$  and notifies Alice
5. (Optional) If they are nearby, Alice and Bob could exchange their exact locations and detect cheating



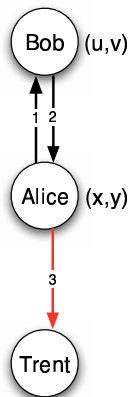
# Louis Protocol

1. Alice sends her encrypted location information to Bob
2. Bob picks a random value  $k$ , and sends  $\mathcal{E}_A(d + k)$ ,  $\mathcal{E}_T(k)$  to Alice
3. Alice sends  $d + k$ ,  $\mathcal{E}_T(k)$  to semi-trusted third party, Trent
4. Trent checks the sign of  $d$  and notifies Alice
5. (Optional) If they are nearby, Alice and Bob could exchange their exact locations and detect cheating



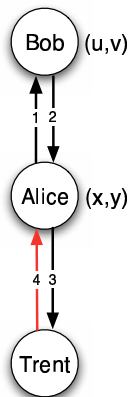
# Louis Protocol

1. Alice sends her encrypted location information to Bob
2. Bob picks a random value  $k$ , and sends  $\mathcal{E}_A(d + k)$ ,  $\mathcal{E}_T(k)$  to Alice
3. Alice sends  $d + k$ ,  $\mathcal{E}_T(k)$  to semi-trusted third party, Trent
4. Trent checks the sign of  $d$  and notifies Alice
5. (Optional) If they are nearby, Alice and Bob could exchange their exact locations and detect cheating



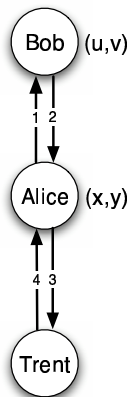
# Louis Protocol

1. Alice sends her encrypted location information to Bob
2. Bob picks a random value  $k$ , and sends  $\mathcal{E}_A(d + k)$ ,  $\mathcal{E}_T(k)$  to Alice
3. Alice sends  $d + k$ ,  $\mathcal{E}_T(k)$  to semi-trusted third party, Trent
4. Trent checks the sign of  $d$  and notifies Alice
5. (Optional) If they are nearby, Alice and Bob could exchange their exact locations and detect cheating



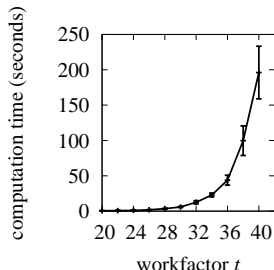
# Louis Protocol

1. Alice sends her encrypted location information to Bob
2. Bob picks a random value  $k$ , and sends  $\mathcal{E}_A(d + k)$ ,  $\mathcal{E}_T(k)$  to Alice
3. Alice sends  $d + k$ ,  $\mathcal{E}_T(k)$  to semi-trusted third party, Trent
4. Trent checks the sign of  $d$  and notifies Alice
5. (Optional) If they are nearby, Alice and Bob could exchange their exact locations and detect cheating



# Lester Protocol

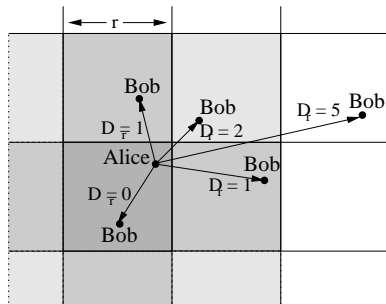
- ▶ No third party
- ▶ Bob chooses a *workfactor*  $t$
- ▶ Using CGS97, Alice needs to solve discrete log in decryption
- ▶ Using Pollard lambda (kangaroo) method, the discrete log can be computed in time  $O(r \cdot 2^{t/2})$  and space  $O(t \log r)$



Alice's computation time in  
 the Lester protocol

## Pierre Protocol: Step One

- ▶ Convert location  $(x, y)$  to coordinates of grid cell  $(x_r, y_r) = (\lfloor \frac{x}{r} \rfloor, \lfloor \frac{y}{r} \rfloor)$
- ▶  $D_r = (x_r - u_r)^2 + (y_r - v_r)^2$  is the square of the distance between Alice and Bob
- ▶ Nearby if  $D_r = 0, 1$  or  $2$



## Pierre Protocol: Step Two

- ▶ Alice  $\rightarrow$  Bob:  $r, \mathcal{E}_A(x_r^2 + y_r^2), \mathcal{E}_A(2x_r), \mathcal{E}_A(2y_r)$
- ▶ Bob picks three random elements  $\rho_0, \rho_1, \rho_2$  of  $\mathbb{Z}_p^*$
- ▶ Bob  $\rightarrow$  Alice:  $\mathcal{E}_A(\rho_0 \cdot D_r), \mathcal{E}_A(\rho_1 \cdot (D_r - 1)), \mathcal{E}_A(\rho_2 \cdot (D_r - 2))$
- ▶ Alice can decrypt them only if  $D_r = 0, D_r - 1 = 0$  or  $D_r - 2 = 0$  respectively
- ▶ How does this work?

## Homomorphic Scheme Used by Pierre Protocol

- ▶ CGS97—A variant of El Gamal
- ▶ Alice's private key is a random element  $a \in \mathbb{Z}_q$  and her public key is  $A = g^a \bmod p$
- ▶ Encryption
  - ▶  $\mathcal{E}(m) = (c_1, c_2) = (g^r \bmod p, A^{r+m} \bmod p)$
- ▶ Decryption
  - ▶  $A^m = c_2 \cdot c_1^{-a} \bmod p$  and compute  $m$  as the discrete log of that value with base  $A \bmod p$
  - ▶ Infeasible unless  $m$  has special properties

## More Details about Pierre Protocol

- ▶  $\mathcal{E}_A(D_r) = (c_1, c_2) = (g^r \bmod p, A^{r+D_r} \bmod p)$
- ▶  $\mathcal{E}_A(\rho_0 \cdot D_r) = (c_1^{\rho_0}, c_2^{\rho_0}) = (g^{r\rho_0} \bmod p, A^{r\rho_0+D_r\rho_0} \bmod p)$
- ▶ To decrypt:  $c_2^{\rho_0} / c_1^{\rho_0 a} = A^{r\rho_0+D_r\rho_0-r\rho_0} \bmod p = A^{\rho_0 D_r}$
- ▶  $\rho_0$  is a random number. Alice can solve the discrete log only if  $D_r = 0$
- ▶ Same technique for  $D_r = 1$  and  $D_r = 2$  or even larger with higher communication cost

## Saving Communication Cost by Ring Homomorphism

- ▶ Both additive and multiplicative
  - ▶ Given  $\mathcal{E}(x)$  and  $\mathcal{E}(y)$ , one can efficiently compute both  $\mathcal{E}(x + y)$  and  $\mathcal{E}(x \cdot y)$
- ▶ Bob could reply with the single ciphertext  $\mathcal{E}_A(\rho \cdot D_r \cdot (D_r - 1) \cdot (D_r - 2))$
- ▶ However, no such secure cryptosystem is yet known to exist
- ▶ Best we can do is to use Boneh-Goh-Nissim:  $\mathcal{E}_A(\rho_1 \cdot D_r \cdot (D_r - 1)), \mathcal{E}_A(\rho_2 \cdot (D_r - 2))$

# Implementation

- ▶ Implemented using the OpenSSL and NTL libraries
- ▶ 2048 bits keys for asymmetric encryption functions
- ▶ Authentication and secure communication:
  - ▶ TLS using AES256 in CBC mode with ephemeral Diffie-Hellman key exchange
- ▶ Tested on 3.0GHz Pentium 4 desktop

# Performance

	Alice	Bob	Trent
TLS connection time	$516 \pm 2$ ms	$255 \pm 4$ ms	$256 \pm 2$ ms
Computation time	$635 \pm 4$ ms	$175 \pm 4$ ms	$41 \pm 0.6$ ms

Runtime of the Louis protocol

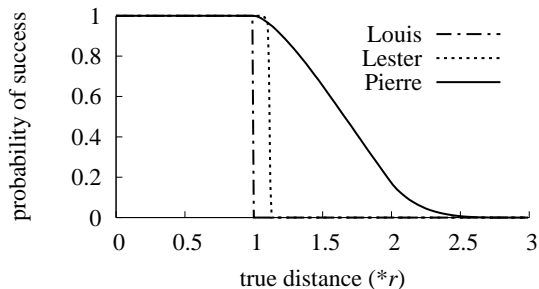
	Alice	Bob
TLS connection time	$256 \pm 3$ ms	$257 \pm 1$ ms
Computation time	$384 \pm 4$ ms	$354 \pm 3$ ms

Runtime of the Pierre protocol

- ▶ Computation time is comparable to setting up a TLS connection

## Comparison: Success Probability

- ▶ Alice *succeeds* if she discovers Bob is nearby



## Comparison: Features

Protocol	Louis	Louis (w/ optional phase)	Lester	Pierre
Extra information learned by Alice	none	Bob's exact location	Bob's exact distance	Bob's grid cell distance
Requires third party	✓	✓		
Bob learns $r$	✓	✓		✓
Bob learns Alice's location		✓		
Communication steps	4	6	2	2

# Conclusion

- ▶ Three protocols to solve the nearby-friend problem
- ▶ Each of them has different features
- ▶ No trusted third party required
- ▶ Feasible communication and computation cost

# Acknowledgments

This work is supported by the Natural Sciences and  
Engineering Research Council of Canada

## More Details about Louis Protocol 1

1. **First phase:** Alice determines her location  $(x, y)$  and her desired radius  $r$ , and picks a random salt  $s_A$ .  
 Alice  $\rightarrow$  Bob:  $\mathcal{E}_A(x^2 + y^2)$ ,  $\mathcal{E}_A(2x)$ ,  $\mathcal{E}_A(2y)$ ,  $r$ ,  $\mathcal{H}(x \parallel y \parallel s_A)$
2. Bob checks the value of  $r$ . If he thinks  $r$  is too large, he aborts the protocol. Otherwise, he determines his location  $(u, v)$ , picks a random value  $k$  and computes

$$\mathcal{E}_A(d + k) = \frac{\mathcal{E}_A(x^2 + y^2) \cdot \mathcal{E}_A(u^2 + v^2) \cdot \mathcal{E}_A(k)}{(\mathcal{E}_A(2x))^u \cdot (\mathcal{E}_A(2y))^v \cdot \mathcal{E}_A(r^2)},$$

Bob also chooses a random salt  $s_B$ .

- Bob  $\rightarrow$  Alice:  $\mathcal{E}_A(d + k)$ ,  $\mathcal{E}_T(k)$ ,  $\mathcal{H}(u \parallel v \parallel s_B)$ ,  $\mathcal{H}(k)$ .
3. Alice decrypts  $\mathcal{E}_A(d + k)$ .  
 Alice  $\rightarrow$  Trent:  $d + k$ ,  $\mathcal{E}_T(k)$ ,  $\text{sig}_A(d + k)$ ,  $\text{sig}_A(\mathcal{E}_T(k))$

## More details about Louis Protocol 2

4. Trent decrypts  $\mathcal{E}_T(k)$  and verifies Alice's signatures. Next, he computes  $d$ . If  $d < 0$ , Trent sets  $answer = 'YES'$  else  $answer = 'NO'$ .  
 Trent  $\rightarrow$  Alice:  $answer$ ,  
 $sig_T(answer \parallel sig_A(d + k) \parallel sig_A(\mathcal{E}_T(k)))$ .
5. Alice verifies Trent's signature. Next, if  $answer == 'YES'$ , she knows that Bob is nearby. Alice terminates the protocol if Bob is not nearby or if only the first phase of the protocol is run. Otherwise:  
**Second phase:** Alice reveals her location to Bob:  
 Alice  $\rightarrow$  Bob:  $answer$ ,  $d + k$ ,  $sig_A(d + k)$ ,  $sig_A(\mathcal{E}_T(k))$ ,  
 $sig_T(answer \parallel sig_A(d + k) \parallel sig_A(\mathcal{E}_T(k)))$ ,  $x$ ,  $y$ ,  $s_A$ .

## More details about Louis Protocol 3

6. Bob verifies all signatures. He then computes  $\mathcal{H}(x \parallel y \parallel s_A)$  and compares the hash value with the one provided by Alice in step 1. He also uses  $(x, y)$  to compute  $d + k$  and compares it to the value received. If the values do not match, Bob aborts the protocol. Otherwise Bob reveals his location to Alice:  
 Bob  $\rightarrow$  Alice:  $u, v, s_B, k$ .
7. Alice computes  $\mathcal{H}(u \parallel v \parallel s_B)$  and  $\mathcal{H}(k)$  and compares the values with the hash values provided by Bob in step 2. Alice also computes  $d + k$  based on  $(x, y)$ ,  $(u, v)$ , and  $k$  and verifies whether it equals the decrypted value of  $\mathcal{E}_A(d + k)$ .

## Security Analysis of Louis Protocol

- ▶ Locations are committed to each other by hash value
- ▶ Parameters used for the encryption are saved
- ▶ Messages are signed to prevent cheating
- ▶ Suspicious behaviour can be detected after the second phase
- ▶ Bob can refuse to answer multiple queries from Alice if they arrive within a very short time (probing attack)
- ▶ However, collusion cannot be detected

## More Details about Lester Protocol

- ▶ Let  $a$  and  $b$  be Alice and Bob's private keys, and  $A = g^a$  and  $B = g^b$  be their public keys
- ▶ Alice and Bob can each calculate  $C = A^b = B^a$
- ▶ Alice  $\rightarrow$  Bob:  $\mathcal{E}_A(x^2 + y^2)$ ,  $\mathcal{E}_A(2x)$ ,  $\mathcal{E}_A(2y)$
- ▶ Bob  $\rightarrow$  Alice:  $t, \mathcal{E}_A(b \cdot (D \cdot 2^t + s))$ 
  - ▶  $D = (x - u)^2 + (y - v)^2$ ,  $t$  is the workfactor chosen by Bob
- ▶ Alice can calculate  $A^{b \cdot (D \cdot 2^t + s)} = C^{D \cdot 2^t + s}$
- ▶ Determining if  $D \cdot 2^t + s$  is in the range  $[0, r^2 \cdot 2^t]$  requires  $O(r \cdot 2^{t/2})$  time and  $O(t \log r)$  space

## Security Analysis of Lester Protocol

- ▶ No way to detect if Alice or Bob use incorrect locations
- ▶ Alice could also check specific range of  $D$
- ▶ Bob can refuse to participate without letting Alice know by encrypting a random value
- ▶ Alice can detect Bob twice far away by using only twice amount of work

# Paillier

- ▶ Alice selects random primes  $p$  and  $q$  and constructs  $n = pq$
- ▶ Alice picks a random  $g \in \mathbb{Z}_{n^2}^*$  and verifies that  $\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n$  exists, where  $\lambda = \text{lcm}(p-1, q-1)$  and  $L(x) = (x-1)/n$
- ▶ Alice's public key is then  $(n, g)$  and her private key is  $(\lambda, \mu)$
- ▶ Encryption:  $c = \mathcal{E}(m) = g^m \cdot r^n \bmod n^2$
- ▶ Decryption:  $\mathcal{D}(c) = L(c^\lambda \bmod n^2) \cdot \mu \bmod n$
- ▶  $\mathcal{E}(m_1 + m_2) = \mathcal{E}(m_1) \cdot \mathcal{E}(m_2) \bmod n^2 = g^{m_1+m_2} \cdot (r_1 r_2)^n \bmod n^2$