The Twin Diffie-Hellman Problem and Applications

David Cash
Georgia Tech

Eike Kiltz
CWI

Victor Shoup
NYU

GEORGIA TECH INFORMATION SECURITY CENTER

(Hashed) ElGamal Encryption

Ingredients:

- **(Enc,Dec)** - Symmetric enc scheme
- **H** - Hash function
- g - generator of G, prime order

$pk = X = g^x$

Pick random y

$Y = g^y$, $K = H(X^y)$

$c = Enc_K(m)$

(Y, c)

$sk = x$

$K = H(Y^x)$

$m = Dec_K(c)$

Proving ElGamal Secure

Necessary for security:

Given random g^x, g^y computing $DH(g^x, g^y) = g^{xy}$ is hard.

This is the Diffie-Hellman assumption.

Claim:

The Diffie-Hellman assumption is not sufficient to prove CCA security!

DH is not sufficient for CCA Security

Consider the following CCA adversary:

pk = X,   given Y, Z:

Choose random m

$K = H(Z)$, $c = Enc_K(m)$

(Y, c) →

$K' = H(Y^x)$

$m' = Dec_{K'}(c)$

← m'

DH is not sufficient for CCA Security

Consider the following CCA adversary:

pk = X,   given Y, Z:

Choose random m

$K = H(Z)$, $c = \mathrm{Enc}_K(m)$

(Y, c) →

$K' = H(Y^x)$

$m' = \mathrm{Dec}_{K'}(c)$

← m'

Case 1: $Z = DH(X,Y)$

Then $m' = m$ always

Case 2: $Z \neq DH(X,Y)$

Then $m' \neq m$ w.h.p.

DH is not sufficient for CCA Security

- A CCA adversary is able to test if DH(**X**, **Y**) = **Z** for **Y** and **Z** of its choosing.
- Thus giving the adversary a decryption oracle also gives him a Decisional DH oracle.
- But evaluating DDH queries is hard for the adversary alone, and thus *some* information about **x** may be leaked by decryption queries.
- How can we prove security of ElGamal?

Stronger Assumptions

Fix a predicate called DHP:

$$DHP(X, Y, Z) = 1 \quad \text{iff} \quad DH(X, Y) = Z$$

Gap DH Assumption [Okamoto, Pointcheval '01]

Hard to compute:

$$DH(g^x, g^y) = g^{xy} \quad \text{with } DHP(\cdot, \cdot, \cdot) \text{ oracle}$$

Strong DH Assumption [Abdalla, Bellare, Rogaway '01]

Hard to compute:

$$DH(g^x, g^y) = g^{xy} \quad \text{with } DHP(g^x, \cdot, \cdot) \text{ oracle}$$

All equivalent to DH assumption in pairing groups, but not in general (?)

Proving Security of ElGamal

Option #1: Use an assumption stronger than DH.

Theorem: [ABR'01] ElGamal is secure against chosen ciphertext attacks in the random oracle model, if

- Strong DH assumption holds
- (Enc, Dec) is chosen ciphertext attack secure

But making stronger assumptions is undesirable.

Proving Security of ElGamal

Option #2: Prove security from the DH assumption, but add some redundancy to the ciphertext.

This is done in all DH-Based schemes: Fujisaki-Okamoto, GEM, REACT, ...

But longer ciphertexts are undesirable for some applications.

New Option: Twin Diffie-Hellman

- Another way to modify ElGamal so that:
    1. We can prove security from the DH assumption
    2. The ciphertext length remains short (like ElGamal).
- This modification is actually a general technique:
    - We define a interactive variant of the Diffie-Hellman problem called the **Strong Twin Diffie-Hellman problem.**
    - We show **Strong Twin Diffie-Hellman assumption** is equivalent to the **(ordinary) Diffie-Hellman** assumption.
    - **Key point:** We give an **interactive** assumption that is equivalent to **(ordinary)** Diffie-Hellman assumption.

More Twinning

- Same technique works for Bilinear and Decisional versions of the DH assumption.
- We give several applications of technique to design schemes with improvements and simple security proofs from well-studied DH assumptions:
    - Encryption - Random Oracle and Standard Model
    - Key exchange
    - Identity Based Encryption (bilinear form)
    - More...

Strong Twin Diffie-Hellman

Twin Diffie-Hellman (2DH) Assumption

Hard to compute:

$2DH(g^{x}, g^{x'}, g^{y}) = (g^{xy}, g^{x'y})$

Define a "twin" predicate called 2DHP:

$2DHP(X, X', Y, Z, Z') = 1$ iff $2DH(X, X', Y) = (Z, Z')$

Strong Twin Diffie-Hellman Assumption

Hard to compute:

$2DH(g^{x}, g^{x'}, g^{y}) = (g^{xy}, g^{x'y})$

w/ $2DHP(g^{x}, g^{x'}, \cdot, \cdot, \cdot)$ oracle

Strong Twin Diffie-Hellman

Tw

H

Theorem:

Strong Twin Diffie Hellman assumption holds iff the Diffie-Hellman assumption holds.

De

Strong Twin Diffie-Hellman Assumption

Hard to compute:

$2DH(g^{x}, g^{x'}, g^{y}) = (g^{xy} , g^{x'y})$

w/ $2DHP(g^{x} , g^{x'}, \cdot , \cdot , \cdot)$ oracle

Proof of Main Theorem

Theorem: Strong 2DH hard ⇔ DH hard

Part 1: Strong 2DH hard ⇒ DH hard (Almost trivial)

Part 2: DH hard ⇒ Strong 2DH hard

How to reduce: outline

1. DH adversary gets (**X**, **Y**) as input.
2. Compute some **X'** related to **X**.
3. Provide strong 2DH adversary with (**X**, **X'**, **Y**) and answer DHP(**X**, **X'**, · , · , ·) oracle queries.
4. Strong 2DH adversary outputs (**Z**, **Z'**), and DH adversary outputs **Z**.

Proof: DH $\Rightarrow$ Strong 2DH

- Assume there exists Strong Twin DH adversary **B**
- Construct **DH** adversary **A:**

  Input: (**X**, **Y**)

  Idea: $X' := g^r X^s$ $(= g^{x'}, x' = r + xs)$

  Run strong 2DH adversary on (**X**, **X'**, **Y**)

  B outputs (**Z**, **Z'**) and **A** returns **Z**.

- How to simulate Strong Twin **DH** adversary's oracle?

  2DHP(X, X', · , · , ·)

  A doesn't know x, x'!

Tool: Trapdoor Test

- Correct answer:

2DHP(**X**, **X'**, **Y**, **Z**, **Z'**) = 1 **iff** "**2DH**(**X**, **X'**, **Y**) = (**Z**, **Z'**)"

iff $X^y = Z$ **and** $X'^y = Z'$

- Simulated answer:

SIM(**X**, **X'**, **Y**, **Z**, **Z'**) = 1 **iff** $Y^r Z^s = Z'$

Claim: Conditioned on any fixed **X'**:

Correct answer = Simulated answer

with prob. **1 - 1/|G|** (over **r**, **s**).

(Proof is simple case analysis)

DH ⇒ Strong 2DH

- If all oracle queries answered correctly, then simulation of Strong 2DH problem is perfect.

B solves Strong 2DH ⇒

A solves DH w.p. **1 - (#queries)/|G|**

- Reduction is tight: reductions to Strong 2DH imply reductions to DH with similar tightness.

Application 1: Twin ElGamal

$pk = (X, X') = (g^{x}, g^{x'})$

Pick random y

$Y = g^{y}$, $K = H(X^{y}, X'^{y})$

$c = Enc_K(m)$

(Y, c)

$sk = (x, x')$

CCA secure if

1. H modeled as random oracle
2. (Enc,Dec) is CCA secure
3. The DH assumption holds

$K = H(Y^{x}, Y^{x'})$

$m = Dec_K(c)$

Twin ElGamal v. Other Schemes

Pros:

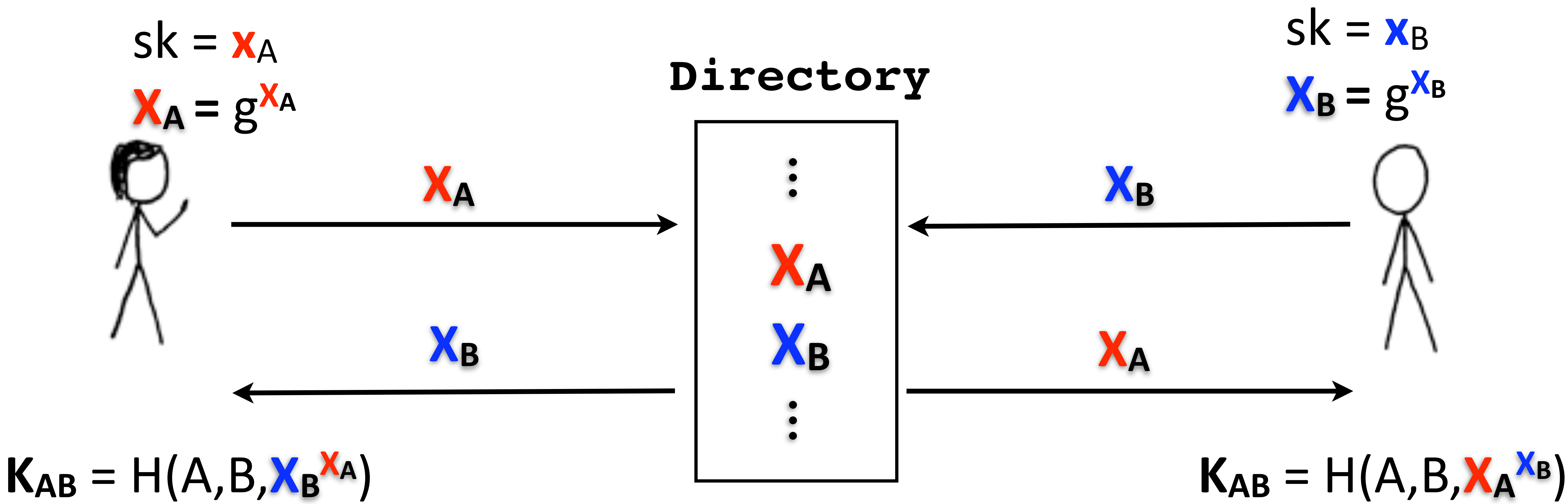
1. Security based on DH, not Strong DH.
2. Short ciphertexts - one group element of overhead when (Enc,Dec) is length-preserving.
3. Analysis is simple - essentially like Hashed ElGamal, except using Strong Twin DH instead of Strong DH.

Cons:

1. Slower encryption (decryption can be optimized though).
2. Larger keys.

Non-Interactive Key Exchange

- All public keys stored in a directory - symmetric keys computed offline
- **Security**: symmetric keys look random to adversary who inserts “rogue keys” into directory.



Non-Interactive Key Exchange

Security of DH protocol:
Secure against active adversaries in random oracle model if the **Strong** DH assumption holds.

sk = x_A
$X_A = g^{x_A}$

$K_{AB} = H(A,B,X_B^{x_A})$

Directory

⋮
X_A
X_B
⋮

sk = x_B
$X_B = g^{x_B}$

$K_{AB} = H(A,B,X_A^{x_B})$

Application 2: Twin DH Key Exchange

Security of Twin DH protocol:

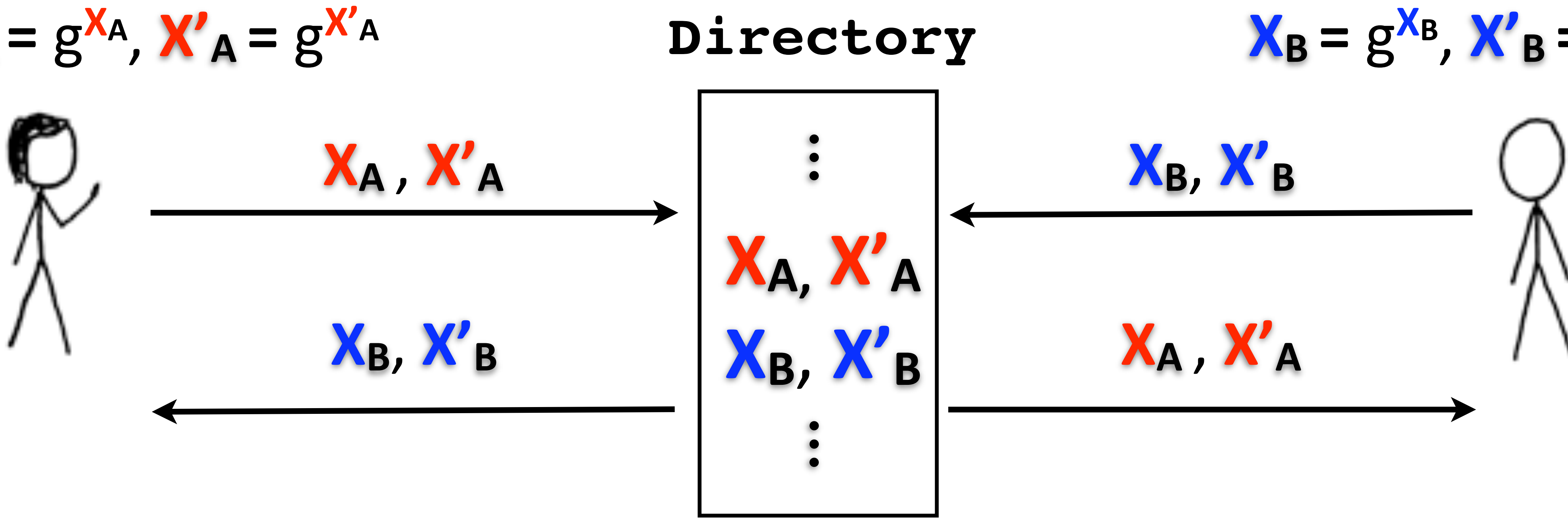
Secure against active adversaries in random oracle model if the DH assumption holds.

sk = x_A, x'_A

$X_A = g^{x_A}$, $X'_A = g^{x'_A}$

sk = x_B, x'_B

$X_B = g^{x_B}$, $X'_B = g^{x'_B}$



$K_{AB} = H(A,B,X_B^{x_A},X_B^{x'_A},X'_B^{x_A},X'_B^{x'_A})$

$K_{AB} = H(A,B,X_A^{x_B},X'_A^{x_B},X_A^{x'_B},X'_A^{x'_B})$

Application 3: Twin Cramer-Shoup

We give a new efficient CCA-secure public-key encryption scheme without random oracles.

- Security based on **Hashed DDH assumption**, which is generally weaker than DDH.
  - Reduce to **Strong Twin Hashed-DDH assumption**, i.e. Hashed DDH with an oracle.
- Simple analysis - resembles some IBE proofs
- Variant gives security from DH assumption (not DDH!), but is less efficient and has a loose reduction.
- Similar unpublished schemes due to [Waters] and [Hanaoka, Kurosawa]

Other applications

1. Identity Based Encryption
   - Twin Boneh-Franklin/Sakai-Kasahara: Short ciphertexts and tighter reduction, but less efficient.
2. Simple technique for securing Password Authenticated Key Exchange against server compromise.
3. Analysis of Shoup's Diffie-Hellman "self corrector".

Conclusion

- General technique: Twin Diffie-Hellman and Trapdoor Test
  - Interactive assumptions that are useful and no stronger than basic DH-type assumptions
- Applications
  1. ElGamal encryption
  2. CCA encryption without random oracles
  3. Non-interactive key exchange
  4. PAKE
  5. IBE
  6. More... see full version on eprint.

Thank you!