

Encryption Beyond Group Homomorphism: Bilinear Groups

Lecture 18

Homomorphic Encryption

- **Group Homomorphism:** Two groups G and G' are homomorphic if there exists a function (homomorphism) $f:G \rightarrow G'$ such that for all $x, y \in G$, $f(x) +_{G'} f(y) = f(x +_G y)$
- **Homomorphic Encryption:** A CPA secure (public-key) encryption s.t. $\text{Dec}(C) +_M \text{Dec}(D) = \text{Dec}(C +_C D)$ for ciphertexts C, D
 - i.e. $\text{Enc}(x) +_C \text{Enc}(y)$ is like $\text{Enc}(x +_M y)$
 - Interesting when $+_C$ doesn't require the decryption key
- e.g., El Gamal: $(g^{x_1}, m_1 Y^{x_1}) \times (g^{x_2}, m_2 Y^{x_2}) = (g^{x_3}, m_1 m_2 Y^{x_3})$
- e.g., Paillier: $g^{m_1} r_1^n \times g^{m_2} r_2^n = g^{m_1+m_2} r_3^n$

Homomorphic Encryption

- **Ring Homomorphism:** Two rings A and A' are homomorphic if there exists a function (homomorphism) $f:A \rightarrow A'$ s.t. $\forall x, y \in A$, $f(x) +_{A'} f(y) = f(x +_A y)$ and $f(x) \times_{A'} f(y) = f(x \times_A y)$
- Fully Homomorphic Encryption: A CPA secure (public-key) encryption s.t. $\text{Enc}(x) +_c \text{Enc}(y)$ is like $\text{Enc}(x +_M y)$ and $\text{Enc}(x) \times_c \text{Enc}(y)$ is like $\text{Enc}(x \times_M y)$
 - Candidate solutions since 2009 using “lattice” problems
 - Today: a simpler kind of encryption, which supports only one multiplication (and any number of additions before and after the multiplication)
 - Uses “bilinear pairings”

Bilinear Pairing

- Two (or three) groups with an efficient pairing operation, $e: G \times G \rightarrow G_T$ that is "bilinear"
 - Typically, prime order (cyclic) groups
 - $e(g^a, g^b) = e(g, g)^{ab}$
 - Multiplication (once) in the exponent!
 - $e(g^a, g^b) e(g^{a'}, g^b) = e(g^{a+a'}, g^b)$; $e(g^a, g^{bc}) = e(g^{ac}, g^b)$; ...
 - Not degenerate: $e(g, g) \neq 1$
- **Decisional Bilinear Diffie-Hellman (DBDH) Assumption:**
For random (a, b, c, z) , the distributions of (g^a, g^b, g^c, g^{abc}) and (g^a, g^b, g^c, g^z) are indistinguishable

3-Party Key Exchange

- A single round 3-party key-exchange protocol secure against passive eavesdroppers (under D-BDH assumption)
 - Generalizes Diffie-Hellman key-exchange
- Let $e: G \times G \rightarrow G_T$ be bilinear and g a generator of G
- Alice broadcasts g^a , Bob broadcasts g^b , and Carol broadcasts g^c
- Each party computes $e(g,g)^{abc}$
 - e.g. Alice computes $e(g,g)^{abc} = e(g^b, g^c)^a$
 - By D-BDH the key $e(g,g)^{abc} = e(g, g^{abc})$ is pseudorandom given eavesdropper's view (g^a, g^b, g^c)

Identity-Based Encryption

- A key-server (with a master secret-key MSK and a master public-key MPK) that can generate $(PK, SK) = (ID, SK_{ID})$ for any given ID (“fancy public-key”)
 - Encryption will use MPK , and the receiver’s ID
 - Receiver has to obtain SK_{ID} from the authority

IBE from Pairing

• MPK: $g, h, Y = e(g, h)^\gamma, \pi = (u, u_1, \dots, u_n)$

• MSK: h^γ

$$\pi(\text{ID}) = u \prod_{i:\text{ID}_i=1} u_i$$

• $\text{Enc}(m; s) = (g^r, \pi(\text{ID})^r, M \cdot Y^r)$

• SK for ID: $(g^t, h^\gamma \cdot \pi(\text{ID})^t) = (d_1, d_2)$

• $\text{Dec}(a, b, c; d_1, d_2) = c / [e(a, d_2) / e(b, d_1)]$

• CPA security based on Decisional-BDH

Some More Assumptions

- **Computational-BDH Assumption:** For random (a,b,c) , given (g^a, g^b, g^c) infeasible to find g^{abc}
- **Decision-Linear Assumption:** $(h_1, h_2, g, h_1^x, h_2^y, g^{x+y})$ and $(h_1, h_2, g, h_1^x, h_2^y, g^z)$ are indistinguishable
- **Strong DH Assumption:** For random x , given (g, g^x) infeasible to find $g^{1/x}$ or even $(y, g^{1/(x+y)})$. (Note: can check $e(g^x g^y, g^{1/(x+y)}) = e(g, g)$.)
 - **q-SDH:** Given (g, g^x, \dots, g^{x^q}) , infeasible to find $(y, g^{1/(x+y)})$
- **Subgroup-Decision Assumption:** Indistinguishability of random elements in G from those in a large subgroup of G (requires G to have composite order)
- **XDDH** when $e: G_1 \times G_2 \rightarrow G_T$: DDH in G_1 and/or G_2 (note: requires $G_1 \neq G_2$)

BGN Encryption

- Boneh–Goh–Nissim Encryption scheme
 - Supports one multiplication and any number of additions through a layer of encryption
 - Based on the **Subgroup-Decision Assumption**
 - $e: G \times G \rightarrow G_T$ where G is a cyclic group with a large non-trivial subgroup
 - $|G| = pq$, a product of two (similar-sized) primes
 - $H \subseteq G$ generated by $h=g^q$, where g generates G , has $|H|=p$
 - Assumption: A random element in H is indistinguishable from a random element in G (cf. DCR)

BGN Encryption

- $e: G \times G \rightarrow G_T$ where G is a cyclic group with $|G|=pq$, and Subgroup-Decision assumption holds for $H \subseteq G$, $|H|=p$ (i.e., $H=\langle g^q \rangle$)
- Message space = Ring of integers modulo n
 - But efficient decryption will be provided only for a small subset of messages
 - In fact, correct decryption will be possible only up to G/H (i.e., $m \in \{0, \dots, q-1\}$) even inefficiently
- Idea: $\text{Enc}_{g,h}(m;r) = g^m h^r$, where g generates G and $h=g^q$ generates H , so that encrypted messages can be added by multiplying ciphertexts, multiplied by plaintext by exponentiating, and **multiplied together by pairing ciphertexts**
 - $e(g^{m+qr}, g^{m'+qr'}) = g_T^{mm' + qr''}$ where $g_T = e(g,g)$ generates G_T

BGN Encryption

- Key generation: Sample $n = pq$, G s.t. $|G|=n$, and generator g for H . Public key includes (G,g,h) and secret-key is (G,g,p) .
- $\text{Enc}_{g,h}(m;r) = g^m h^r$, where g generates G and $h=g^q$ generates H
- $\text{Dec}_{g,p}(c)$: Find m s.t. $g^m p = c^p$ (by brute force, when m is from a small set)
 - $c^p = g^{mp} h^{rp} = g^{mp}$ since $h^p = g^q = 1$
- Homomorphic operations (in group G):
 - $c_1 +_C c_2 = c_1 \cdot c_2$, $a *_C c = c^a$ and $c_1 \times_C c_2 = e(c_1, c_2)$. $\text{rerand}(c) = c \cdot h^r$.
 - But \times_C results in a ciphertext in G_T ! Decryption, homomorphic addition and multiplication by plaintext (but not multiplication of two encrypted values), rerand defined for these ciphertexts too
- CPA secure under Subgroup-Decision assumption on G and H (which implies the same for G_T and H_T): Encryption using a random element in G instead of h^r (random element in H) has no information about message.

Quadratic speedup using "Pollard's Kangaroo method" for discrete log

2-DNF Computation using BGN Encryption

- Consider a passive-secure 2-party computation problem where Bob has an input bit-vector x and Alice has a secret "2-DNF formula" f . Bob should get $f(x)$ only, and Alice should learn nothing.
 - Disjunctive Normal Form: OR (disjunction) of ANDs
 - 2-DNF: $\bigvee_{i=1}^n (y_i \wedge z_i)$ where y_i, z_i are literals (input variables or their negations)
 - Passive-secure protocol:
 - Bob generates keys for BGN encryption, encrypts each bit using it, and sends the PK and ciphertexts to Alice
 - Alice homomorphically computes $c \leftarrow \text{Enc}(r \cdot f'(x))$ where f' is a degree-2 polynomial version of f , using $+$ for \vee and \times for \wedge and $(1-x)$ for $\neg x$, and r random. Bob can (only) check if $f'(x)=0$ or not.

Full-fledged decryption not needed in the protocol

2-DNF Computation using BGN Encryption

- In some applications, want to protect against encryption of illegal values
- Suppose we require $m \in \{0,1\}$. But BGN allows $m \in \{0, \dots, q-1\}$.
- Can protect against revealing information by blinding encrypted outputs
 - Instead of returning a ciphertext c , return $c +_c \text{Enc}(\alpha)$, where $\alpha=0$ if all given values are valid, and random otherwise
 - $\alpha = \sum_{i=1 \text{ to } n} r_i \cdot x_i \cdot (1-x_i)$
 - In BGN, $\text{Enc}(\alpha)$ (in G_T) can be computed from $\{ \text{Enc}(x_i) \}_i$ (in G)

Beyond One Multiplication?

- Instead of bilinear maps, if n -linear maps are available, can support up to degree n polynomials
 - Open problem to construct good candidates for multi-linear maps
- Fully Homomorphic Encryption: No a priori bound on the degree of the polynomials that can be homomorphically evaluated. Polynomial may be specified as an arithmetic circuit
- Levelled Homomorphic Encryption
 - Homomorphic encryption supporting any polynomial a priori upper bound on the (mult.) depth of the circuit to be evaluated
 - Ciphertexts of different levels, based on number of mult. used
- Somewhat Homomorphic Encryption: Works only for some functions (e.g., log-depth circuits)