CS 276 – Cryptography Sept 8, 2014

Lecture 18: Witness Encryption

Instructor: Sanjam Garg *Scribe: Fotis Iliopoulos*

1 A story

Imagine that a billionaire who loves mathematics, would like to award with 1 million dollars the mathematician(s) who will prove the Riemann Hypothesis. Of course, neither does the billionaire know if the Riemann Hypothesis is true, nor if he will be still alive (if and) when a mathematician will come up with a proof. To overcome these couple of problems, the billionaire decides to:

1. Put 1 million dollars in gold in a big treasure chest.
2. Choose an arbitrary place of the world, dig up a hole, and hide the treasure chest.
3. Encrypt the coordinates of the treasure chest in a message so that only the mathematician(s) who can actually prove the Riemann Hypothesis can decrypt it.
4. Publish the ciphertext in every newspaper in the world.

The goal of this lecture is to help the billionaire with step 3. To do so, we will assume for simplicity that the proof is at most 10000 pages long. The latter assumption implies that the language

$$L = \{x \text{ such that } x \text{ is an acceptable Riemann Hypothesis proof}\}$$

is in NP and therefore, using a reduction, we can come up with a circuit C that takes as input x and outputs 1 if x is a proof for the Riemann Hypothesis or 0 otherwise.
Our goal now is to design a pair of PPT machines (Enc, Dec) such that:

1. $\text{Enc}(C, m)$ takes as input the circuit C and $m \in \{0,1\}$ and outputs a ciphertext $e \in \{0,1\}^*$.
2. $\text{Dec}(C, e, w)$ takes as input the circuit C, the cipertext e and a witness $w \in \{0,1\}^*$ and outputs m if if $C(w) = 1$ or $\perp$ otherwise.

and so that they satisfy the following correctness and security requirements:

- **Correctness:** If $\exists w$ such that $C(w) = 1$ then $\text{Dec}(C, e, w)$ outputs m.
- **Security:** If $\nexists w$ such that $C(w) = 1$ then $\text{Enc}(C, 0) \approx^c \text{Enc}(C, 1)$ (where $\approx^c$ means "computationally indistinguishable").

2 A Simple Language

As a first example, we show how we can design such an encryption scheme for a simple language. Let G be a group of prime order and g be a generator of the group. For elements $A, B, T \in G$ consider the language $L = \{(a, b) : A = g^a, B = g^b, T = g^{ab}\}$. An encryption scheme for that language with the correctness and security requirements of Section 1 is the following:

- **Encryption**(g, A, B, T, G)**:**
  - Choose elements $r_1, r_2 \in \mathbb{Z}_p^*$ uniformly and independently.
  - Let $c_1 = A^{r_1} g^{r_2}$, $c_2 = g^m T^{r_1} B^{r_2}$, where $m \in \{0, 1\}$ is the message we want to encrypt.
  - Output $c = (c_1, c_2)$
- **Decryption(**b**):**
  - Output $\frac{c_2}{c_1^b}$

Correctness: The correcntess of the above encryption scheme follows from the fact that if there exist $(a, b) \in L$ then:

$$\begin{aligned} \frac{c_2}{c_1^b} &= \frac{g^m T^{r_1} B^{r_2}}{(A^{r_1} g^{r_2})^b} \\ &= \frac{g^m \left(g^{ab}\right)^{r_1} \left(g^b\right)^{r_2}}{(g^a)^{r_1 b}\, g^{r_2 b}} \\ &= g^m \end{aligned}$$

Since $m \in \{0, 1\}$ and we know g, the value of g^m implies the value of m.

Security: As far as the security of the scheme is concerned, since L is quite simple, we can actually prove that m is information-theoreticaly hidden. To see this, assume there does not exist $(a, b) \in L$, but an adversary has the power to compute discrete logarithms. In that case, given c_1 and c_2 the adversary could get a system of the form:

$$\begin{aligned} ar_1 + r_2 &= s_1 \\ m + rr_1 + br_2 &= s_2 \end{aligned}$$

, where s_1 and s_2 are the discrete logarithms of c_1 and c_2 respectively (with base g), and $r \neq ab$ is an element of $\mathbb{Z}_p^*$ such that $T = g^r$. Observe now that for each value of m there exist numbers r_1 and r_2 so that the above system has a solution, and thus m is indeed information-theoretically hidden (on the other hand, if we had that $ab = r$ then the equations are linearly dependent).

3 An NP Complete Language

In this section we focus to our original goal of designing an encryption for an NP complete language L. Specifically, we will consider the NP-complete problem *exact cover*. Besides that, we introduce the n-Multilinear Decisional Diffie-Hellman (n-MDDH) assumption and the Decision Multilinear No-Exact-Cover Assumption (see also [1]). The latter will guarantee the security of our construction.

3.1 Exact Cover

We are given as input $x = (n, S_1, S_2, \ldots, S_l)$, where n is an integer and each $S_i, i \in [l]$ is a subset of $[n]$, and our goal is to find a subset of indices $T \subseteq [l]$ such that:

1. $\cup_{i \in T} S_i = [n]$ and
2. $\forall i, j \in T$ such that $i \neq j$ we have that $S_i \cap S_j \neq \emptyset$.

If such a T exists, we say that T is an exact cover of x.

3.2 Multilinear Maps

Mutlinear maps is a generalization of bilinear maps (which we have already seen) that will be useful in our construction. Specifically (following the definition of []), we assume the existence of a group generator $\mathcal{G}$, which takes as input a security parameter n and a positive integer n to indicate the number of allowed operations. $\mathcal{G}(1^\lambda, n)$ outputs a sequence of groups $\vec{\mathbb{G}} = (\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_n)$ each of large prime order $P > 2^\lambda$. In addition, we let g_i be a canonical generator of $\mathbb{G}_i$ (and is known from the group's description).
We also assume the existence of a set of bilinear maps $\{e_{i,j} : \mathbb{G}_i \times \mathbb{G}_j \to \mathbb{G}_{i+j} \mid i, j \geq 1; i + j \leq n\}$. The map $e_{i,j}$ satisfies the following relation:

$$e_{i,j}\left(g_i^a, g_j^b\right) = g_{i+j}^{ab} : \forall a, b \in \mathbb{Z}_p \tag{1}$$

and we observe that one consequence of this is that $e_{i,j}(g_i, g_j) = g_{i+j}$ for each valid i, j.

3.3 The n-MDDH Assumption

The n- Multilinear Decisional Diffie-Hellman (n-MDDH) problem states the following: A challenger runs $\mathcal{G}(1^\lambda, n)$ to generate groups and generators of order p. Then it picks random $s, c_1, \ldots, c_n \in \mathbb{Z}_p$. The assumption then states that given $g = g_1, g^s, g^{c_1}, \ldots, g^{c_n}$ it is hard to distinguish $T = g_n^{s \prod_{j \in [1,n]} c_j}$ from a random group element in G_n, with better than negligible advantage (in security parameter λ).

3.4 Decision Multilinear No-Exact-Cover Assumption

Let $x = (n, S_1, \dots, S_l)$ be an instance of the exact cover problem that has no solution. Let param $\leftarrow \mathcal{G}(1^{1+n}, n)$ be a description of a multilinear group family with order $p = p(\lambda)$. Let $a_1, a_2, \dots, a_n, r$ be uniformly random in $\mathbb{Z}_p$. For $i \in [l]$, let $c_i = g_{|S_i|}^{\prod_{j \in S_i} a_j}$. Distiguish between the two distributions:

$$(\text{params}, c_1, \dots, c_l, g_n^{a_1 a_2 \dots a_n}) \text{ and } (\text{params}, c_1, \dots, c_l, g_n^r)$$

The Decision Multilinear No-Exact-Cover Assumption is that for all adversaries $\mathcal{A}$, there exists a fixed negligible function $\nu()$ such that for all instances x with no solution, $\mathcal{A}$'s distinguishing advantage against the Decision Multilinear No-Exact-Cover Problem for x is at most $\nu(\lambda)$.

3.5 The Encryption Scheme

We are now ready to give the description of our encryption scheme.

- Enc(x, m) takes as input $x = (n, S_1, \dots, S_l)$ and the message $m \in \{0, 1\}$ and:
  - Samples $a_0, a_1, \dots, a_n$ uniformly and independently from $\mathbb{Z}_p^*$.
  - $\forall i \in [l]$ let $c_i = g_{|S_i|}^{\prod_{j \in S_j} a_j}$
  - Sample uniformly an element $r \in \mathbb{Z}_p^*$
  - Let $d = d(m)$ be $g_n^{\prod_{j \in [n]} a_j}$ if $m = 1$ or g_n^r if $m = 0$.
  - Output $c = (d, c_1, \dots, c_l)$
- Dec(x, T), where $T \subseteq [l]$ is a set of indices, computes $\prod_{i \in T} c_i$ and outputs 1 if the latter value equals to d or 0 otherwise.
- **Correctness:** Assume that T is an exact cover of x. Then, it is not hard to see that:

$$\begin{aligned} \prod_{i \in T} c_i &= \prod_{i \in T} g_{|S_i|}^{\prod_{j \in S_j} a_j} \\ &= g_n^{\prod_{j \in [n]} a_j} \end{aligned}$$

  , where we have used (1) repeatededly and the fact that T is an exact cover (to show that $\sum_{i \in T} |S_i| = n$ and that $\prod_{i \in T} \prod_{j \in S_i} = \prod_{i \in [n]} a_i$).

- **Security:** Intuitively, the construction is secure, since the only way to make $g_n^{\prod_{j \in [n]} a_i}$ is to find an exact cover of $[n]$. As a matter of fact, observe that if an exact cover does not exist, then for each subset of indices T'(such that $\cup_{i \in T'} S_j = [n]$) we have that

$$\sum_{i=1}^{n} |S_i| > n$$

  , which means that $\prod_{i \in T} \prod_{j \in S_i} a_i$ is different than $\prod_{j \in [n]} a_j$. Formally, the security is based on the Decision Multilinear No-Exact-Cover Assumption.

References

[1] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 467–476. ACM, 2013.