Introduction to Modern Cryptography
Recitation 13

Orit Moskovich
Tel Aviv University
January 25, 2017

Example 1

- Let f_1, f_2 be OWFs
- Is $F(x) = \big(f_1(x), f_2(x)\big)$ necessarily a OWF?

- No!
- Let g be a OWF, and define $f_1(x_1, x_2) = g(x_1), x_2$ ➔ f_1 is a OWF
- Similarly, $f_2(x_1, x_2) = x_1, g(x_2)$ ➔ f_2 is a OWF

- $F(x) = F(x_1, x_2) = \big(f_1(x), f_2(x)\big) = \big(g(x_1), x_2, x_1, g(x_2)\big)$ ➔ not a OWF

- Still need to prove why f_1, f_2 are OWFs

Example 2

- Consider the following key-exchange protocol:
  - Alice chooses $k, r \leftarrow \{0,1\}^n$ at random, and sends $s \coloneqq k \oplus r$ to Bob
  - Bob chooses $t \leftarrow \{0,1\}^n$ at random and sends $u \coloneqq s \oplus t$ to Alice
  - Alice computes $w \coloneqq u \oplus r$ and sends w to Bob
  - Alice outputs k and Bob computes $w \oplus t$

a) Show that Alice and Bob output the same key

b) Show that the scheme is not secure
(Reminder) Secrecy: Given the public information and all the communication exchanged during the execution of the protocol, computing the shared key is computationally hard.

Example 2

- Alice outputs k
- Bob outputs:

$$w \oplus t = (u \oplus r) \oplus t = \big((s \oplus t) \oplus r\big) \oplus t = \Big(\big((k \oplus r) \oplus t\big) \oplus r\Big) \oplus t = k$$

Example 2

- The scheme is not secure.
- Given a transcript (s, u, w) of the protocol, an adversary can compute:

$$s \oplus u \oplus w = (k \oplus r) \oplus u \oplus (u \oplus r) = k$$

Example 3

משרד התקשורת הציע להפעיל ***שירות לתיאום מפתחות הצפנה*** (להלן *שלמה*) שיפעל בצורה הבאה. שלמה בוחר זוג ראשוניים גדולים מאד p, q ומפרסם את מכפלתם $N = pq$. כשאליס ובוב רוצים לתאם מפתח משותף, כל אחד מהם מגריל מספר $1 < r < N$ מקרי – אליס את r_A ובוב את r_B – מעלה אותו בחזקת $e = 3$ מודולו n ושולח לשלמה. שלמה מקבל את $r_A^3 \bmod N$ ואת $r_B^3 \bmod N$, מפענח אותם, ושולח לאליס ולבוב את $r_A + r_B \bmod N$. כעת אליס ובוב יודעים שניהם גם את r_A וגם את r_B ויכולים לחשב מפתח משותף $K = \mathrm{AES}_{r_A}(r_B)$.

כדי להבטיח הגנה מלאה למשתמשים, שלמה שומר אצלו מאגר ביומטרי עם כל השאילתות שנשלחו אליו אי פעם ומסרב לענות פעמיים על אותה שאילתה (כלומר: אם, למשל, שולחים אליו את $r_A^3 \bmod N$ פעם נוספת, הוא מחזיר שגיאה).

מנחם המאזין שמע את r_A^3 ואת r_B^3 ורוצה לחשב את K. הסבירו כיצד הוא ושותפתו למזימה, שפרה, יכולים לנצל את שלמה למטרה זו.

Example 3

- Menachem can choose a random $k \neq 0,1, k \in Z_N$
- He can send $k^3 \cdot r_A^3 \; mod \; M$ to Shlomo
- Shifra sends $k^3 \; mod \; N$ to Shlomo
- Shlomo sends back $k \cdot r_A + k$ from which it is easy to compute r_A

Example 4

- A Sudoku game is a $n \times n$ board partially filled out with numbers $1 \ldots n$

	9			8		4		
		2		4	1			5
3							6	
	1							
7	6			2			1	9
							8	
	2							8
5			2	9		3		
		4		5			2	

- The goal is to fill out the rest of the board with numbers $1 \ldots n$ such that every row, column and the sub-boxes all have exactly one of each digit in them

Example 4

- Consider the following ZK proof between a prover P that holds a solution Sol to a verifier V:

1. $P \to V$: Chooses a random permutation $\sigma: [n] \to [n]$ and sends to V a commitment $c = COM\left(\sigma(Sol)\right)$
2. $V \to P$: Picks at random row/column/sub-box
3. $P \to V$: Reveals the commitment to the cells
4. V accepts iff the values of the cells are different

a) Show soundness and completeness

Example 4

- Consider better ZK proof between a prover P that holds a solution Sol to a verifier V:

1. $P \rightarrow V$: Chooses a random permutation $\sigma: [n] \rightarrow [n]$ and sends to V a commitment $c = COM(\sigma(Sol))$
2. $V \rightarrow P$: Flips a coin $b \in \{0,1\}$
   $b = 0$ ➔ Picks at random row/column/sub-box
   $b = 1$ ➔ Asks for the commitment to the known values on the board
3. $P \rightarrow V$: Reveals the requested commitment
4. V accepts iff the values of the cells are different/a valid permutation

b) Show soundness and completeness

Example 4

- Consider better ZK proof between a prover P that holds a solution Sol to a verifier V:

1. $P \rightarrow V$: Chooses a random permutation $\sigma: [n] \rightarrow [n]$ and sends to V a commitment $c = COM\big(\sigma(Sol)\big)$
2. $V \rightarrow P$: Flips a coin $b \in \{0,1\}$
   $b = 0$ ➔ Picks at random row/column/sub-box
   $b = 1$ ➔ Asks for the commitment to the known values on the board
3. $P \rightarrow V$: Reveals the requested commitment
4. V accepts iff the values of the cells are different/a valid permutation

c) Show a simulator

<u>*Example 5*</u>

- Let COM_1, COM_2 be two commitment schemes
- Both schemes are <u>binding</u>
- However, one of them is not <u>hiding</u>
- To solve the problem, one constructed a new commitment scheme:
- $COM(M) = COM_1(M), COM_2(M)$

- Is COM secure?
- No!
- Let $COM_1(M) = M$ and COM_2 some secure (hiding, binding) scheme
- $COM(M) = M, COM_2(M)$ ➔ not hiding

Example 6

- Let $p = 3 \; mod \; 4$ prime
- Let $a \in QR(Z_P^*)$
- Show that $a^{\frac{p+1}{4}}$ is a square root of a

- Let g be a generator, $a = g^{2i} \; mod \; p$
- $\left(a^{\frac{p+1}{4}}\right)^2 = a^{\frac{p+1}{2}} = g^{2i\left(\frac{p+1}{2}\right)} = g^{i(p-1)+2i} = g^{2i} = a \; mod \; p$

- Finding a square root over Z_p^* is easy for any prime p

Example 7

- Let $p = 3\ mod\ 4$ prime
- Let g be a generator in Z_p^*
- Define the following problems:
  - **Mult**: given (p, g, g^x, g^y) ➔ compute g^{xy}
  - **Square**: given (p, g, g^x) ➔ compute g^{x^2}

- Let A_{mult} be an algorithm that given (p, g, g^x, g^y) returns g^{xy} w.p 1
- Show an algorithm A_{square} that given (p, g, g^x) returns g^{x^2} w.p 1

Example 7

- Let $p = 3 \, mod \, 4$ prime
- Let g be a generator in Z_p^*
- Define the following problems:
  - **Mult**: given (p, g, g^x, g^y) ➔ compute g^{xy}
  - **Square**: given (p, g, g^x) ➔ compute g^{x^2}

- A_{square} that is given (p, g, g^x):
  1. Run A_{mult} on (p, g, g^x, g^x) and get $g^{x \cdot x} = g^{x^2} \, mod \, p$

Example 7

- Let $p = 3\ mod\ 4$ prime
- Let g be a generator in Z_p^*
- Define the following problems:
  - **Mult**: given (p, g, g^x, g^y) ➔ compute g^{xy}
  - **Square**: given (p, g, g^x) ➔ compute g^{x^2}

- Let A_{square} be an algorithm that given (p, g, g^x) returns g^{x^2} w.p 1
- Show an algorithm A_{mult} that given (p, g, g^x, g^y) returns g^{xy} w.p 1

Example 7

- Let $p = 3 \ mod \ 4$ prime
- Let g be a generator in Z_p^*
- Define the following problems:
  - **Mult**: given (p, g, g^x, g^y) ➔ compute g^{xy}
  - **Square**: given (p, g, g^x) ➔ compute g^{x^2}

- A_{mult} that is given (p, g, g^x, g^y):
  1. Run A_{square} on (p, g, g^x) and get $g^{x^2} \ mod \ p$
  2. Run A_{square} on (p, g, g^y) and get $g^{y^2} \ mod \ p$
  3. Run A_{square} on (p, g, g^{x+y}) and get $g^{(x+y)^2} \ mod \ p$
  4. Compute $\frac{g^{(x+y)^2}}{g^{x^2} \cdot g^{y^2}} = g^{2xy}$ and find its square root