CS 6601 Secure Data Analysis - Fall 2017
Homework 1

Adam Bowers
Sammie Bush
Dalton Cole

September 22, 2017

Problem 1 Show how to construct the garbled circuit.

The normal circuit for constructing an "=" operator can be seen in Figure 1. A circuit labeled with wires can be seen in 2. To construct a garbled circuit, first each wire has to be assigned two random t bit strings. There are two strings to cover the 0 and 1 input cases. In an actual garbled circuit, t would be set to 80. For demonstrative purposes, t will be 8 in this case. Table 1 shows the random strings assigned to each wire where v_a^b such that a is the wire and b is the bit the string represents. Next, a permutation bit must be randomly chosen for each wire. This can be found in Table 2. The random permutation is appended to v_i^j to form w_i^j such that $w_k^0 = v_k^0 \| (0 \oplus p_k)$ and $w_k^1 = v_k^1 \| (1 \oplus p_k)$. Each w_i^j value can be seen in Table 3.

Following this, each truth table is replaced with it's corresponding Garbled-Truth-Table (GTT) by replacing each 0 or 1 with w_k^0 or w_k^1 respectively. The GTT is replaced by the Encrypted-Garbled-Truth-Table (EGTT) which is an encrypted version of the GTT. The encryption is performed by SHA1 hashing $v_i^x \| k \| x' \| y'$ with the plane text w_k^x where $x' = x \oplus p_i$ and $y' = y \oplus p_j$ and x, y are the entries in the original truth table. This encryption will be represented as $Enc(w_i^j)$.

To make the Permutted-Encrypted-Garbled-Truth-Table (PEGTT), the rows in the EGTT have to be swapped based on the following rule: if $p_i = 1$, the first two entries of the table are swapped with the last two entries; if $p_j = 1$, then the first and third are swapped and the second and fourth entries are swapped. The GTT, EGTT, and PEGTT for gate 0 can be found in Table 4. For gates 1-6, see Tables 5, 6, 7, 8, 9, 10. For another example, see attached.

Problem 2 Show how to evaluate the garbed circuit securely.

To evaluate the garbled circuit, Bob sends to Alice the garbled circuit along with his wired input, in this case w_0^x, w_2^x, w_4^x, and w_6^x. For Alice to obtain her garbled input (w_1^y, w_3^y, w_5^y, and w_7^y), she must use oblivious transfer (OT) with Bob.

For each gate, the following must take place. Alice splits w_i^x into v_i and x, where x equals the least significant bit of W_i^x and v_i is the remainder of W_i^x. Similarly, split W_j^y into v_j and y, where y equals the least significant bit of W_j^y and v_j is the remainder of w_j^y. The values of x and y are used to index into the PEGTT where x is the high bit and y is the low bit. To retrieve w_k, Alice has to hash $v_i \| k \| x \| y$ and $v_j \| k \| x \| y$, and then XOR the two hashes with the ciphertext.

Alice then uses the transition table she got from Bob to interpret the circuit's true output. She sends the garbled value to Bob, and Bob maps the garbled value to the true value as well.

For example, let's say Bob has a value of 1011 and Alice has a value of 1010. The following tables illistrates how to evaulate the garbled circuit securely: 11, 12, 13, 14, 15, 16, 17.

Figure 1: Normal Circuit

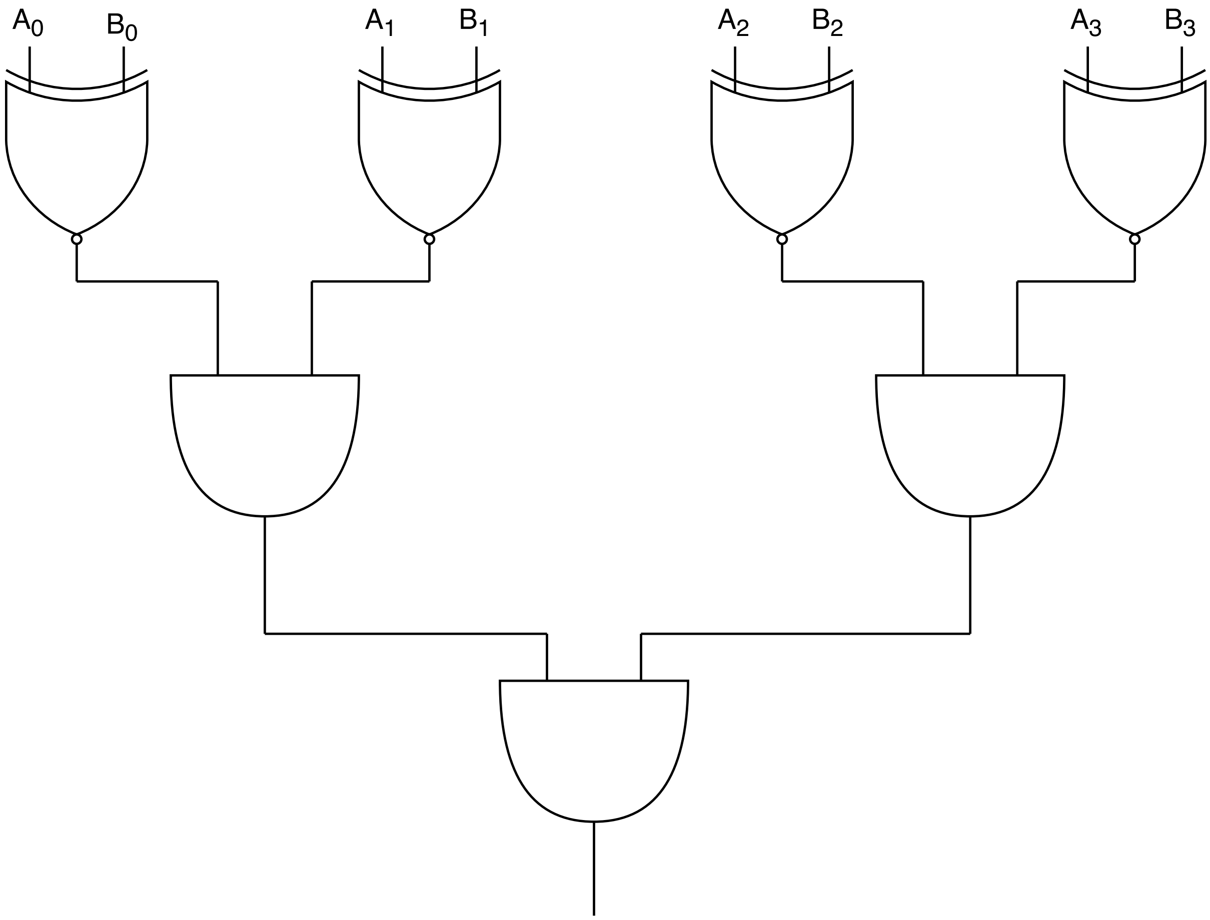

Table 1: Random t-bit Strings

$v_0^0 = 00000000$	$v_0^1 = 11111111$
$v_1^0 = 00000001$	$v_1^1 = 10000000$
$v_2^0 = 00000010$	$v_2^1 = 01000000$
$v_3^0 = 00000011$	$v_3^1 = 11000000$
$v_4^0 = 00000100$	$v_4^1 = 00100000$
$v_5^0 = 00000101$	$v_5^1 = 10100000$
$v_6^0 = 00000110$	$v_6^1 = 01100000$
$v_7^0 = 00000111$	$v_7^1 = 11100000$
$v_8^0 = 00001000$	$v_8^1 = 00010000$
$v_9^0 = 00001001$	$v_9^1 = 10010000$
$v_{10}^0 = 00001010$	$v_{10}^1 = 01010000$
$v_{11}^0 = 00001011$	$v_{11}^1 = 11010000$
$v_{12}^0 = 00001100$	$v_{12}^1 = 00110000$
$v_{13}^0 = 00001101$	$v_{13}^1 = 10110000$
$v_{14}^0 = 00001110$	$v_{14}^1 = 01110000$

Figure 2: Garbled Circuit Wires

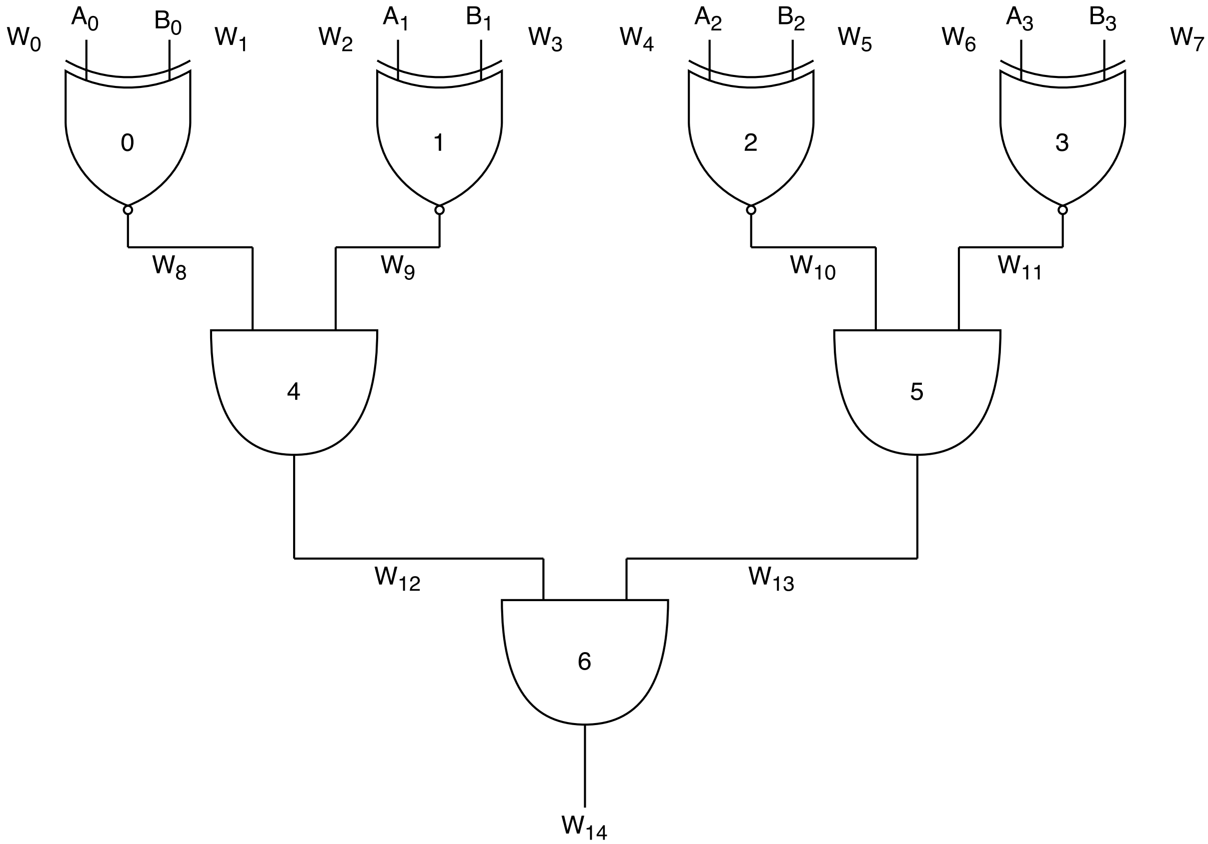

Table 2: Permutation Bit

$p_0 = 0$
$p_1 = 1$
$p_2 = 0$
$p_3 = 1$
$p_4 = 0$
$p_5 = 1$
$p_6 = 0$
$p_7 = 1$
$p_8 = 0$
$p_9 = 1$
$p_{10} = 0$
$p_{11} = 1$
$p_{12} = 0$
$p_{13} = 1$
$p_{14} = 0$

Table 3: Wire Strings

$w_0^0 = 000000000$	$w_0^1 = 111111111$
$w_1^0 = 000000011$	$w_1^1 = 100000000$
$w_2^0 = 000000100$	$w_2^1 = 010000001$
$w_3^0 = 000000111$	$w_3^1 = 110000000$
$w_4^0 = 000001000$	$w_4^1 = 001000001$
$w_5^0 = 000001011$	$w_5^1 = 101000000$
$w_6^0 = 000001100$	$w_6^1 = 011000001$
$w_7^0 = 000001111$	$w_7^1 = 111000000$
$w_8^0 = 000010000$	$w_8^1 = 000100001$
$w_9^0 = 000010011$	$w_9^1 = 100100000$
$w_{10}^0 = 000010100$	$w_{10}^1 = 010100001$
$w_{11}^0 = 000010111$	$w_{11}^1 = 110100000$
$w_{12}^0 = 000011000$	$w_{12}^1 = 001100001$
$w_{13}^0 = 000011011$	$w_{13}^1 = 101100000$
$w_{14}^0 = 000011100$	$w_{14}^1 = 011100001$

Table 4: GTT, EGTT, PEGTT for Gate 0

Truth Table			GTT			EGTT			PEGTT		
X	Y	Out	X	Y	Out	X	Y	Out	X	Y	Out
0	0	1	w_0^0	w_1^0	w_8^1	$Enc(w_0^0)$	$Enc(w_1^0)$	$Enc(w_8^1)$	$Enc(w_0^0)$	$Enc(w_1^1)$	$Enc(w_8^0)$
0	1	0	w_0^0	w_1^1	w_8^0	$Enc(w_0^0)$	$Enc(w_1^1)$	$Enc(w_8^0)$	$Enc(w_0^0)$	$Enc(w_1^0)$	$Enc(w_8^1)$
1	0	0	w_0^1	w_1^0	w_8^0	$Enc(w_0^1)$	$Enc(w_1^0)$	$Enc(w_8^0)$	$Enc(w_0^1)$	$Enc(w_1^1)$	$Enc(w_8^1)$
1	1	1	w_0^1	w_1^1	w_8^1	$Enc(w_0^1)$	$Enc(w_1^1)$	$Enc(w_8^1)$	$Enc(w_0^1)$	$Enc(w_1^0)$	$Enc(w_8^0)$

Table 5: GTT, EGTT, PEGTT for Gate 1

Truth Table			GTT			EGTT			PEGTT		
X	Y	Out	X	Y	Out	X	Y	Out	X	Y	Out
0	0	1	w_2^0	w_3^0	w_9^1	$Enc(w_2^0)$	$Enc(w_3^0)$	$Enc(w_9^1)$	$Enc(w_2^0)$	$Enc(w_3^1)$	$Enc(w_9^0)$
0	1	0	w_2^0	w_3^1	w_9^0	$Enc(w_2^0)$	$Enc(w_3^1)$	$Enc(w_9^0)$	$Enc(w_2^0)$	$Enc(w_3^0)$	$Enc(w_9^1)$
1	0	0	w_2^1	w_3^0	w_9^0	$Enc(w_2^1)$	$Enc(w_3^0)$	$Enc(w_9^0)$	$Enc(w_2^1)$	$Enc(w_3^1)$	$Enc(w_9^1)$
1	1	1	w_2^1	w_3^1	w_9^1	$Enc(w_2^1)$	$Enc(w_3^1)$	$Enc(w_9^1)$	$Enc(w_2^1)$	$Enc(w_3^0)$	$Enc(w_9^0)$

Table 6: GTT, EGTT, PEGTT for Gate 2

Truth Table			GTT			EGTT			PEGTT		
X	Y	Out	X	Y	Out	X	Y	Out	X	Y	Out
0	0	1	w_4^0	w_5^0	w_{10}^1	$Enc(w_4^0)$	$Enc(w_5^0)$	$Enc(w_{10}^1)$	$Enc(w_4^0)$	$Enc(w_5^1)$	$Enc(w_{10}^0)$
0	1	0	w_4^0	w_5^1	w_{10}^0	$Enc(w_4^0)$	$Enc(w_5^1)$	$Enc(w_{10}^0)$	$Enc(w_4^0)$	$Enc(w_5^0)$	$Enc(w_{10}^1)$
1	0	0	w_4^1	w_5^0	w_{10}^0	$Enc(w_4^1)$	$Enc(w_5^0)$	$Enc(w_{10}^0)$	$Enc(w_4^1)$	$Enc(w_5^1)$	$Enc(w_{10}^1)$
1	1	1	w_4^1	w_5^1	w_{10}^1	$Enc(w_4^1)$	$Enc(w_5^1)$	$Enc(w_{10}^1)$	$Enc(w_4^1)$	$Enc(w_5^0)$	$Enc(w_{10}^0)$

Table 7: GTT, EGTT, PEGTT for Gate 3

Truth Table			GTT			EGTT			PEGTT		
X	Y	Out	X	Y	Out	X	Y	Out	X	Y	Out
0	0	1	w_6^0	w_7^0	w_{11}^1	$Enc(w_6^0)$	$Enc(w_7^0)$	$Enc(w_{11}^1)$	$Enc(w_6^0)$	$Enc(w_7^1)$	$Enc(w_{11}^0)$
0	1	0	w_6^0	w_7^1	w_{11}^0	$Enc(w_6^0)$	$Enc(w_7^1)$	$Enc(w_{11}^0)$	$Enc(w_6^0)$	$Enc(w_7^0)$	$Enc(w_{11}^1)$
1	0	0	w_6^1	w_7^0	w_{11}^0	$Enc(w_6^1)$	$Enc(w_7^0)$	$Enc(w_{11}^0)$	$Enc(w_6^1)$	$Enc(w_7^1)$	$Enc(w_{11}^1)$
1	1	1	w_6^1	w_7^1	w_{11}^1	$Enc(w_6^1)$	$Enc(w_7^1)$	$Enc(w_{11}^1)$	$Enc(w_6^1)$	$Enc(w_7^0)$	$Enc(w_{11}^0)$

Table 8: GTT, EGTT, PEGTT for Gate 4

Truth Table			GTT			EGTT			PEGTT		
X	Y	Out	X	Y	Out	X	Y	Out	X	Y	Out
0	0	0	w_8^0	w_9^0	w_{12}^0	$Enc(w_8^0)$	$Enc(w_9^0)$	$Enc(w_{12}^0)$	$Enc(w_8^0)$	$Enc(w_9^1)$	$Enc(w_{12}^0)$
0	1	0	w_8^0	w_9^1	w_{12}^0	$Enc(w_8^0)$	$Enc(w_9^1)$	$Enc(w_{12}^0)$	$Enc(w_8^0)$	$Enc(w_9^0)$	$Enc(w_{12}^0)$
1	0	0	w_8^1	w_9^0	w_{12}^0	$Enc(w_8^1)$	$Enc(w_9^0)$	$Enc(w_{12}^0)$	$Enc(w_8^1)$	$Enc(w_9^1)$	$Enc(w_{12}^1)$
1	1	1	w_8^1	w_9^1	w_{12}^1	$Enc(w_8^1)$	$Enc(w_9^1)$	$Enc(w_{12}^1)$	$Enc(w_8^1)$	$Enc(w_9^0)$	$Enc(w_{12}^0)$

Table 9: GTT, EGTT, PEGTT for Gate 5

Truth Table			GTT			EGTT			PEGTT		
X	Y	Out	X	Y	Out	X	Y	Out	X	Y	Out
0	0	0	w_{10}^0	w_{11}^0	w_{13}^0	$Enc(w_{10}^0)$	$Enc(w_{11}^0)$	$Enc(w_{13}^0)$	$Enc(w_{10}^0)$	$Enc(w_{11}^1)$	$Enc(w_{13}^0)$
0	1	0	w_{10}^0	w_{11}^1	w_{13}^0	$Enc(w_{10}^0)$	$Enc(w_{11}^1)$	$Enc(w_{13}^0)$	$Enc(w_{10}^0)$	$Enc(w_{11}^0)$	$Enc(w_{13}^0)$
1	0	0	w_{10}^1	w_{11}^0	w_{13}^0	$Enc(w_{10}^1)$	$Enc(w_{11}^0)$	$Enc(w_{13}^0)$	$Enc(w_{10}^1)$	$Enc(w_{11}^1)$	$Enc(w_{13}^1)$
1	1	1	w_{10}^1	w_{11}^1	w_{13}^1	$Enc(w_{10}^1)$	$Enc(w_{11}^1)$	$Enc(w_{13}^1)$	$Enc(w_{10}^1)$	$Enc(w_{11}^0)$	$Enc(w_{13}^0)$

Table 10: GTT, EGTT, PEGTT for Gate 6

Truth Table			GTT			EGTT			PEGTT		
X	Y	Out	X	Y	Out	X	Y	Out	X	Y	Out
0	0	0	w_{12}^0	w_{13}^0	w_{14}^0	$Enc(w_{12}^0)$	$Enc(w_{13}^0)$	$Enc(w_{14}^0)$	$Enc(w_{12}^0)$	$Enc(w_{13}^1)$	$Enc(w_{14}^0)$
0	1	0	w_{12}^0	w_{13}^1	w_{14}^0	$Enc(w_{12}^0)$	$Enc(w_{13}^1)$	$Enc(w_{14}^0)$	$Enc(w_{12}^0)$	$Enc(w_{13}^0)$	$Enc(w_{14}^0)$
1	0	0	w_{12}^1	w_{13}^0	w_{14}^0	$Enc(w_{12}^1)$	$Enc(w_{13}^0)$	$Enc(w_{14}^0)$	$Enc(w_{12}^1)$	$Enc(w_{13}^1)$	$Enc(w_{14}^1)$
1	1	1	w_{12}^1	w_{13}^1	w_{14}^1	$Enc(w_{12}^1)$	$Enc(w_{13}^1)$	$Enc(w_{14}^1)$	$Enc(w_{12}^1)$	$Enc(w_{13}^0)$	$Enc(w_{14}^0)$

Table 11: Decrypting Gate 0

w_0		w_1	
$v_0 = 11111111$	$x = 1$	$v_1 = 10000000$	$y = 0$
$w_8 = hash(v_0\|8\|x\|y) \oplus hash(v_1\|8\|x\|y) \oplus Enc(w_8^1)$			
Transition Table$(w_8^1) = 1$			

Table 12: Decrypting Gate 1

w_2		w_3	
$v_2 = 10000000$	$x = 0$	$v_3 = 00000011$	$y = 1$
$w_9 = hash(v_2\|9\|x\|y) \oplus hash(v_3\|9\|x\|y) \oplus Enc(w_9^1)$			
Transition Table$(w_9^1) = 1$			

Table 13: Decrypting Gate 2

w_4		w_5	
$v_4 = 001000001$	$x = 1$	$v_5 = 101000000$	$y = 0$
$w_{10} = hash(v_4\|10\|x\|y) \oplus hash(v_5\|10\|x\|y) \oplus Enc(w_{10}^1)$			
Transition Table$(w_{10}^1) = 1$			

Table 14: Decrypting Gate 3

w_6		w_7	
$v_6 = 01100000$	$x = 1$	$v_7 = 11100000$	$y = 0$
$w_{11} = hash(v_6\|11\|x\|y) \oplus hash(v_7\|11\|x\|y) \oplus Enc(w_{11}^0)$			
Transition Table$(w_{11}^0) = 0$			

Table 15: Decrypting Gate 4

w_8		w_9	
$v_8 = 00010000$	$x = 1$	$v_9 = 10010000$	$y = 0$
$w_{12} = hash(v_8\|12\|x\|y) \oplus hash(v_9\|12\|x\|y) \oplus Enc(w_{12}^0)$			
Transition Table$(w_{12}^1) = 1$			

Table 16: Decrypting Gate 5

w_{10}		w_{11}	
$v_{10} = 01010000$	$x = 1$	$v_{11} = 00001011$	$y = 1$
$w_{13} = hash(v_{10}\|13\|x\|y) \oplus hash(v_{11}\|13\|x\|y) \oplus Enc(w_{13}^0)$			
Transition Table$(w_{13}^0) = 0$			

Problem 3 Show how the evaluator knows which entry in the garbled circuit can be decrypted correctly.

In a 1-2 OT, Bob has two messages m_0 and m_1 and Alice has a bit b. Alice wants to retrieve m_b, without Bob knowing b.

1. Bob sends N, e, x_0, and x_1 to Alice, i.e. x_0, x_1 are randomly chosen from $1, ..., N-1$. (Bob also knows the private key d)
2. Alice randomly selects $k \in 1, ..., N-1$
3. Alice sends $v = (k^e modN) \oplus x_b$ to Bob
4. Bob computes $k_0 = (v \oplus x_0)^d modN = k$. $k_1 = (v \oplus x_1)^d modN = k$, i.e. $k^e \oplus x_0 \oplus x_0 \Rightarrow k$ and $k^e \oplus x_1 \oplus x_0 \Rightarrow k$
5. Bob sends $z_0 = m_0 \oplus k_0 = m_0 \oplus k$ and $z_1 = m_1 \oplus k_1 = m_0 \oplus k$ to Alice
6. Alice computes $m_b = z_b \oplus k$, i.e. Alice is only able to retrieve one value of either x_0 or x_1 while knowing k

Problem 4 What is the appropriate key size for constructing the garbled circuit?

Fairplay - A Secure Two-Party Computation System used 80 bits for the key size. However that alone doesnt mean the key size is appropriate. The key needs to be large enough to firstly, mask a given input to the circuit. Given the examples in the paper only used inputs of 1 bit, this isnt an issue for those circuits. The other concern when deciding a key size is to make brute force guessing of the keys or other attacks impractical. Now the necessary size to combat this is debatable and always changing with advancements in hardware and theory, but in regards to papers given 80 bits should suffice.

Table 17: Decrypting Gate 6

w_{12}		w_{13}	
$v_{12} = 00110000$	$x = 1$	$v_{13} = 00001101$	$y = 1$
$w_{14} = hash(v_{12}\|14\|x\|y) \oplus hash(v_{13}\|14\|x\|y) \oplus Enc(w_{14}^0)$			
Transition Table$(w_{14}^0) = 0$			