IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON SECURITY AND PRIVACY FOR CLOUD AND IOT

Received February 6, 2019, accepted February 17, 2019, date of publication February 21, 2019, date of current version April 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900519

Privacy-Preserving Wildcards Pattern Matching Protocol for IoT Applications

HONG QIN, HAO WANG, XIAOCHAO WEI, LIKUN XUE, AND LEI WU
School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China

Corresponding author: Hao Wang (wanghao@sdnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602287, Grant 61802235, Grant 61672330, and Grant 61572294, in part by the Primary Research and Development Plan of Shandong Province under Grant 2018GGX101037, and in part by the Major Innovation Project of Science and Technology of Shandong Province under Grant 2018CXGC0702.

ABSTRACT With the continuous development of the Internet of Things (IoT), various IoT devices create an incomprehensible amount of data all the time. However, the IoT devices have limited computing and storage resources and are difficult to process massive data locally, so they often introduce servers to help them for calculating or analyzing data. At present, the ''IoT + Cloud'' mode has been widely accepted. How to protect users' privacy in the public cloud environment has become critical. Among the common methods of processing data in the server, pattern matching is an important one which aims to identify the appearance and location of a string (called pattern) within a larger string or text. There are a lot of studies on privacy-preserving pattern matching protocols, but most protocols are constructed using heavy public-key cryptographic operations, which are not applicable to IoT devices. In this paper, we propose a new protocol using secret sharing and oblivious transfer (OT) and latter improve its efficiency with OT extension, so it is very efficient for lightweight IoT devices. In addition, our protocol also supports query with wildcards which can be used for the batch search. This protocol is provable-secure against honest-but-curious adversaries. Both the theoretical and experimental results show that our protocol can be used in real-world IoT applications.

INDEX TERMS Privacy-preserving, wildcards pattern matching, secret sharing, oblivious transfer, Internet of Things.

I. INTRODUCTION

The Internet of Things (IoT) is an important part of the new generation of information technology. Its core and foundation is still the Internet but is an extended network and the clients in IoT extend to any things. IoT is widely used in real-world applications through communication-aware technologies such as intelligent sensing, identification technology and pervasive computing. It is also considered to be the third wave of the development of world information industry after Computer and Internet. Recently, according to Cisco and Ericsson's predictions, more than 20 billion IoT devices will be connected to the Internet by 2021 [1]. Therefore, IoT industry is considered to be one of the most promising industries in the future.

In many application scenarios, the IoT technology connects a large number of sensors to network to collect real-time data, and it combines sensing capabilities with computational and data analysis capabilities of back-end applications to extract valuable information. In recent years, with the development of IoT devices becoming more and more perfect, the data collected by device also shows diversification of uses, such as smart cities, smart medical and industrial internet of things etc. In smart city scene, the sensor collects data about traffic, energy, air quality and other real-time data and uploads it to the back-end server [2]–[4]. After server processes the data, it immediately gives feedback and the IoT devices make response. In smart medical scene, the wearable device collects patient's body data in real time and uploads it to the medical server which processes data and provides feedback [5]–[8]. Then the wearable device displays different content based on feedback and doctor can infer patient's health condition. In industrial IoT scene, there are a huge amount of real-time data in the process of production. Due to the limited computing power of device, it is necessary to introduce a cloud platform to analyze the data and return the

The associate editor coordinating the review of this manuscript and approving it for publication was Weizhi Meng.

extracted useful information to devices [9], [10]. The information is propitious to make decision about production for company.

In these applications, when it needs to determine whether the data collected by IoT devices satisfies certain features, pattern matching as a basic technology in computer science is often needed. It's essentially a search problem that finds the position of a given pattern $p \in [\sum]^m$ in the text $t \in [\sum]^n$, where Σ is an alphabet set. However, in the distributed computing scenario, the leakage of private information is becoming more and more serious. People do not want to disclose their own information when performing pattern matching. Therefore, it is necessary to ensure that the private data is not leaked. The research on secure pattern matching protocol can be traced back to [11] in 2007. They focus on exact pattern matching and turn them into evaluation problems of oblivious automata. Later, there are few references [12]–[16] to improve its efficiency and security. In addition, some scholars use different techniques such as oblivious pseudo-random computing [17], [18] and Yao garbled circuits [19] to give different protocol structures.

In recent years, the development of pattern matching has been mainly manifested in the functional extension, including approximate pattern matching, wildcard pattern matching and so on. Research on privacy-preserving wildcards pattern matching is a hot topic in recent years. Hazay and Toft [20], [21] convert wildcard pattern matching to exact pattern matching using additive homomorphic encryption scheme. Baron *et al.* [22] studied the generalization of non-binary alphabets. Their main idea is based on linear algebraic formulas and additive homomorphic encryption scheme. In addition, a wildcard pattern matching protocol based on symmetric somewhat homomorphic encryption scheme are constructed in [23] and [24]. They constructed data packaging method that efficiently calculates multiple Hamming distances of encrypted data. Their protocol can be applied to non-binary data and can query 16,500-length gene sequence per second. In 2017, Kolesnikov *et al.* [25] constructed an efficient secure wildcard pattern matching protocol based on OT extension. They use OT protocol so that two participants can calculate a random value together and then invoke secure string equality test (SSET) to determine whether the pattern matching is successful. Recently, Darivandpour and Atallah [26] give a more efficient protocol construction. Their scheme fits all character sets and supports input of any size.

As we all know, the data generated by IoT devices often acts as a pattern and to match specific text holding by server. Then the interaction result is returned to device and device responds differently according to it. Taking smart medical as an example. The sensor collects physiological parameters of the patient and transmits them to corresponding sever by wireless communication, then the sever matches the received data with the text to obtain patient's health status or disease information. This process is shown in Fig.1. In order to protect the privacy of both parties, device shouldn't send

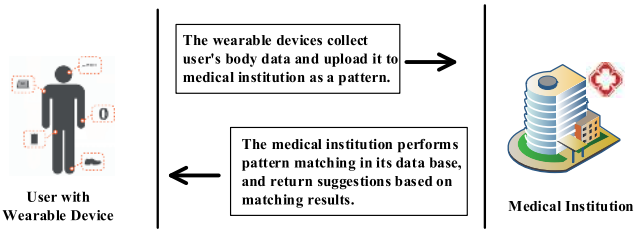

FIGURE 1. Pattern matching in smart medical.

the information collected by sensor directly and the server shouldn't obtain final result of the matching problem. So a privacy-preserving pattern matching protocol is needed. The protocol should output location where pattern appears in text while satisfying the following security attributes: (1) pattern p is kept secret to server; (2) the pattern provider doesn't know anything else in the text other than where p appears. There are a few references about privacy-preserving pattern matching protocols as we discussed above, but most of them mainly use public-key cryptographic operation which are difficult to run on IoT devices. Therefore, we propose a lightweight privacy-preserving wildcards pattern matching protocol in this paper, so that the device providing pattern and the server owning text can complete matching problem without leaking their own data.

In this paper, we propose an efficient wildcards pattern matching protocol. Specifically, we first compose a protocol using secret sharing and oblivious transfer and latterly improve its efficiency with OT extension. The offline phase of our protocol only requires XOR operations on bit strings without defining other data structures, so it is very efficient and suitable for lightweight IoT devices. We prove its security against semi-honest adversaries. Both theoretical and experiment show that our scheme is capable of real-world applications.

The rest of the paper is organized in the following manner. In section II, we introduce some preliminaries and definitions. Then, we propose the construction of our two-party wildcard pattern matching protocol and give its correctness analysis in section III. In section IV, we give the security proof in the semi-honest model. In section V, experiments and application of IoT are shown. At the end, we present the conclusion of our work in section VI.

II. PRELIMINARIES AND DEFINITIONS

A. SECRET SHARING

Secret sharing scheme was first proposed by Shamir [27] and Blakley [28] in 1979. Its function is to distribute a secret to multiple participants, each of whom gets a share of the secret. The secret can be reconstructed only if the number of shares exceeds the threshold. In this paper, we use a trivial secret sharing scheme, called XOR-secret-sharing scheme. It is an (n, n)-threshold secret sharing scheme. That is to say, the secret is divided into n shares, and all shares are needed

Inputs:
- The sender S inputs (x_0, x_1);
- The receiver R inputs a choice bit $b \in \{0, 1\}$.

Outputs:
- The receiver R outputs x_b ;
- The sender S outputs nothing.

FIGURE 2. The oblivious transfer functionality $\mathcal{F}_{OT_2^1}$.

Inputs:
- The sender S inputs $x_0 \in \{0, 1\}^*$;
- The receiver R inputs $x_1 \in \{0, 1\}^*$.

Outputs:
- The receiver R outputs 1 if $x_0 = x_1$ and 0 otherwise;
- The sender S outputs nothing.

FIGURE 3. The secure string equality test functionality $\mathcal{F}_{SSET}$.

to reconstruct the secret. There are two algorithms in this scheme:

- **Secret splitting algorithm:**
  It takes secret $s \in \{0, 1\}^\lambda$ and number of shares m as input, and sets m secret shares in the following way, where λ is the security parameter:
  For $i \in [1, m-1]$, it selects $s_i \in \{0, 1\}^\lambda$ randomly, and calculates $s_m = s \oplus s_1 \oplus s_2 \oplus \ldots \oplus s_{m-1}$.
  It outputs $s_1, s_2, \ldots, s_m$ as secret shares.
- **Secret reconstruction algorithm:**
  It takes m secret shares $s_1, s_2, \ldots, s_m$ as input, and output secret s as $s = s_1 \oplus s_2 \oplus \ldots \oplus s_m$.

We note that the secret s can be reconstructed if and only if all m shares are correct.

B. OBLIVIOUS TRANSFER

The oblivious transfer (OT) protocol was first proposed by Rabin [29] in 1981. As a basic protocol in cryptography, it has been widely used in secure multiparty computation. OT protocol involves two parties, one is a sender S and the other is a receiver R. Sender S transfers a set of messages to receiver R, and receiver R can obtain a subset of the messages, but sender S does not know what messages he received. In 1-out-of-2 OT (OT_2^1), which we used in this paper, a sender S with inputs x_0 and x_1 interacts with a receiver R who has a input choice bit $b \in \{0, 1\}$. After that, the receiver R gets the output x_b without learning anything about x_{1-b}. The sender S has no output and learns nothing about b. In the following, we give a detailed description about the functionality of OT_2^1 in Fig.2:

C. SECURE STRING EQUALITY TEST

To our knowledge, secure string equality test protocol was first proposed by Fagin *et al.* [30] in 1996. There are two party in this protocol, the sender S holds a string x_0, and the receiver R holds a string x_1. At the end of the protocol, R learns whether $x_0 = x_1$ and nothing else, while S learns nothing. The functionality of it is proposed in Fig.3.

D. COMPUTATIONAL INDISTINGUISHABILITY

Let $X = \{X(a, n)\}_{a\in\{0,1\}^*;n\in\mathbb{N}}$ and $Y = \{Y(a, n)\}_{a\in\{0,1\}^*;n\in\mathbb{N}}$ be two distribution ensembles indexed by a security parameter n; we say X and Y are computationally indistinguishable, i.e. $X \overset{c}{\equiv} Y$, if for any probabilistic polynomial time (PPT) algorithm $\mathcal{A}$ with input $a \in \{0, 1\}^*$ and $n \in \mathbb{N}$, the following quantity is a negligible function in n:

$$|\mathbf{Pr}[\mathcal{A}(X(a, n)) = 1] - \mathbf{Pr}[\mathcal{A}(Y(a, n)) = 1]| \leq \varepsilon(n).$$

E. SECURITY DEFINITION

We mainly consider the semi-honest adversary, that is, two participants strictly follow the protocol, but expect to obtain input information of the other from their own view. Our formal definitions here are according to [31]. We present a formalization based on the simulation paradigm.

- Let f: $\{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}^* \times \{0, 1\}^*$ be a probabilistic polynomial-time functionality and π be a two-party protocol for computing f. $f_1(x, y)$ and $f_2(x, y)$ represent the first and second elements of $f(x, y)$, respectively.
- In an execution of π on (x,y) and security parameter n, the *view* of the i-th party ($i \in \{1, 2\}$) is denoted by $view_i^\pi(x, y, n)$, that is $(w, r^i, m_1^i, \ldots, m_t^i)$, where $w \in (x, y)$, r^i indicates the content of the i-th party's internal random tape, and m_j^i represents the j-th message that is received.
- In an execution of π on (x,y) and security parameter n, the *output* of the i-th party is denoted by $output_i^\pi(x, y, n)$ and is implicit in the party's view of the execution.

Definition 1: We say π securely computes a functionality f in the presence of static semi-honest adversaries if there exist PPT algorithms S_1 and S_2, such that

$$\{S_1(x, f_1(x, y), f_2(x, y))\} \overset{c}{\equiv} \{View_1^\pi(x, y), output_2^\pi(x, y)\}$$
$$\{(f_1(x, y), S_1(y, f_2(x, y)))\} \overset{c}{\equiv} \{View_2^\pi(x, y), output_1^\pi(x, y)\}$$

where $x, y \in \{0, 1\}^*$.

This definition states that the view of a party can be simulated by a PPT algorithm given access to the party's input and output only.

III. PRIVACY-PRESERVING TWO-PARTY WILDCARD PATTERN MATCHING PROTOCOL

The functionality of two-party wildcard pattern matching $\mathcal{F}_{WPM}$ mainly involves two participants, P_1 and P_2,

Inputs:

- P_1 inputs a text $t \in \{0,1\}^n$ and an integer m;
- P_2 inputs a pattern $p \in \{0,1,*\}^m$ which includes wildcards and an integer n.

Outputs:

- P_1 outputs nothing;
- P_2 outputs the positions if p appears in t and nothing otherwise.

FIGURE 4. The secure two-party wildcard pattern matching functionality $\mathcal{F}_{WPM}$.

in which P_1 holds a text string t and P_2 holds a pattern string p. P_2 wants to get the locations if p appears in t. Again, P_1 and P_2 do not want to reveal their own data (except input length) to each other when performing pattern matching. The functionality is provided as follows in Fig.4.

In order to achieve the functionality $\mathcal{F}_{WPM}$, we propose a privacy-preserving two-party wildcard pattern matching protocol π_{WPM}. First of all, we introduce the protocol at a high level. In order to search a specific m-bit substring in an n-bit main string ($n \geq m$), it needs to slide the substring in the main string bit by bit. An n-bit main string has $n-m+1$ m-bit substrings, therefore, the protocol needs to process $n-m+1$ times. For the k-th time, P_1 holds an m-bit substring t_k, which is the k-th substring of t from the left, and P_2 holds m-bit pattern string p. If and only if t_k is equal to p bit by bit, the matching is successful at location k. In order to test whether t_k is equal to p while protecting the data privacy of both parties, we use secret sharing and oblivious transfer techniques. First of all, P_2 divides a random secret s into m parts according to the length of p, i.e. each bit of p corresponds to a secret share s_i. At the same time, a random share r_i is selected for each bit of p. Then, P_1 and P_2 execute m times 1-out-of-2 OT protocol, which P_1 acts as the receiver and P_2 acts as the sender. In the j-th OT execution, the input of P_1 is the j-th bit of t_k and the input of P_2 is a pair (s_j^0, s_j^1). If the j-th bit of p is 0, set $s_j^0 = s_j$, $s_j^1 = r_j$, else if the j-th bit of p is 1, set $s_j^0 = r_j$, $s_j^1 = s_j$, else the j-th bit of p is wildcard $*$, set $s_j^0 = s_j^1 = s_j$. That is to say, the input of P_2 is either a pair of true share and random share or a pair of same true shares. If the j-th bit of t_k is equal to the j-th bit of p, or the j-th bit of p is wildcard $*$, P_1 will get the true share. After executing m times OT_2^1, if t_k is match to p bit by bit, then P_1 can get all the correct secret shares and recover the secret s. We can judge whether t_k and p are equal by judging whether the secret recovered by P_1 is equal to the secret randomly selected by P_2. In this process, the data privacy of both parties is guaranteed. We note, when a bit in p is a wildcard bit, no matter what the corresponding bit in t_k is, P_1 can always get the corresponding true share.

In the following, we give the full description.

Privacy-preserving two-party wildcard pattern matching protocol π_{WPM}

- **Inputs:**
  - P_1 holds a text $t \in \{0,1\}^n$ and an integer m;
  - P_2 holds a pattern $p \in \{0,1,*\}^m$ and an integer n.

  Let $N = n - m + 1$ represent the number of m-bit substrings in n-bit main string. ($n \geq m$)
- **Input representation phase:**
  For $k \in [1, N]$:
  - P_2 chooses a secret $s_k \in \{0,1\}^\lambda$ randomly, where λ is security parameter. It runs secret splitting algorithm of XOR-secret-sharing scheme to get m secret shares, i.e. $s_{k,1}$, $s_{k,2}, \ldots, s_{k,m} \in \{0,1\}^\lambda$. We have $s_k = s_{k,1} \oplus s_{k,2} \oplus \ldots \oplus s_{k,m}$.
  - Then, P_2 selects m random shares $r_{k,i} \in \{0,1\}^\lambda$, for $i \in [1, m]$.
  - If the i-th bit of p is 0, P_2 sets $(s_{k,i}^0, s_{k,i}^1) = (s_{k,i}, r_{k,i})$, else if the i-th bit of p is 1, P_2 sets $(s_{k,i}^0, s_{k,i}^1) = (r_{k,i}, s_{k,i})$. Otherwise, if the i-th bit of p is $*$, P_2 sets $(s_{k,i}^0, s_{k,i}^1) = (s_{k,i}, s_{k,i})$.

$$\begin{pmatrix} (s_{1,1}^0, s_{1,1}^1) & \cdots & (s_{1,m}^0, s_{1,m}^1) \\ (s_{2,1}^0, s_{2,1}^1) & \cdots & (s_{2,m}^0, s_{2,m}^1) \\ \vdots & \ddots & \vdots \\ (s_{N,1}^0, s_{N,1}^1) & \cdots & (s_{N,m}^0, s_{N,m}^1) \end{pmatrix}$$

  These values are used as inputs of P_2 in the oblivious transfer protocol of next phase.
- **Oblivious transfer and secret reconstruction phase:**
  For $k \in [1, N]$, P_1 and P_2 jointly perform 1-out-of-2 OT protocols m times, where P_1 acts as the receiver and P_2 acts as the sender. In the j-th OT execution:
  - P_1 takes the j-th bit $t_{k,j}$ of substring t_k as input.
  - P_2 takes pair $(s_{k,j}^0, s_{k,j}^1)$ as input.

  After executing OT protocols m times, P_1 gets $s_{k,1}^{t_{k,1}}$, $s_{k,2}^{t_{k,2}}, \ldots, s_{k,m}^{t_{k,m}}$. Then, it reconstructs the secret $s'_k = s_{k,1}^{t_{k,1}} \oplus s_{k,2}^{t_{k,2}} \oplus \ldots \oplus s_{k,m}^{t_{k,m}}$.
- **Output phase:** For $k \in [1, N]$, P_1 and P_2 jointly perform a string equality test protocol. P_1 acts as sender and P_2 acts as receiver. If the reconstructed secret s'_k is equal to s_k, P_2 outputs 1 indicating that the substring t_k and the pattern p match successfully. Then, P_2 outputs k as the position.

Correctness: Before proving the security of this protocol, we firstly analyze the correctness, that is, P_2 will eventually get the correct result. We explain the correctness from the following two aspects:

- If the match is successful, it means that at least one m-bit substring in t matches the pattern p. All values on the non-wildcard bits in pattern p are equal to the values of substring in corresponding positions. Therefore, in OT protocol, all legal secret shares are received by P_1 when this substring is used as input. As for wildcard bits, P_1 always get legal share in corresponding position. Finally, with these legal secret shares, P_1 can reconstruct the same secret s'_k as P_2 randomly selected s_k before, and the output of string equality test with inputs s'_k and s_k must be 1. Thus, P_2 learns that this substring can match successfully with pattern p and knows the starting location of matching substring.
- If there is no successful match, it means that no m-bit substring in t match the pattern p. It indicates that at least one bit on non-wildcard bits in pattern p differs from the value of substring in corresponding position. Therefore, in OT protocol, the value P_1 gets is an unlegal share at this position. According to the functionality of XOR-secret-sharing scheme, the number of legal shares is less than m, which makes it impossible to correctly construct secret s_k. Therefore, the result of the string equality test must be 0, i.e., the match is unsuccessful.

IV. SECURITY ANALYSIS

We prove the security of π_{WPM} in semi-honest adversary model. The protocol mainly involves three cryptographic primitives, namely secret sharing, oblivious transfer and string equality test. Intuitively, since oblivious transfer protocol is secure, the input information of P_1 is kept secret to P_2 which ensures that t is not leaked. According to the nature of secret sharing scheme, it does not reveal the information of shares when the number of shares is insufficient to reconstruct secret. In addition, in string equality test, the participant P_2 with input string p can only receive output 1 or output 0 and knows nothing about P_1. The participant P_1 with input string t does not know anything about p either. In this way, the security of this protocol can be guaranteed through the nature of three basic primitives above.

In the following, we present formal security proof of protocol π_{WPM} based on security definition in previous section.

Theorem 1: If the security of OT, secret sharing and secure string equality test is satisfied, then protocol π_{WPM} securely computes the functionality $\mathcal{F}_{WPM}$ in the presence of semi-honest adversaries.

Proof: We give this proof in a hybrid model where the OT protocol is computed by the ideal functionality $\mathcal{F}_{OT_2^1}$ and the string equality test is computed by the ideal functionality $\mathcal{F}_{SSET}$. The proof contains two separate cases that P_1 is corrupted and P_2 is corrupted.

P_1 Is Corrupted: In an execution of π_{WPM}, P_1's view consists of its view in OT and secure string equality test protocol.

We construct a simulator S_1 with inputs of a text $t \in \{0,1\}^n$ and an integer m and generates the view of P_1 in π_{WPM}. S_1 randomly selects a pattern p to generate secret sharing shares which are transferred to $\mathcal{F}_{OT_2^1}$.

Let S_1^{OT} be the simulator that is used for party P_1 to get its view in the OT protocol. Simulator S_1 invokes the input and output of simulator S_1^{OT} with the purpose of obtaining P_1's view, that is (t, s^σ) in which t is the text and s^σ is secret sharing share ($\sigma \in \{0,1\}$).

Let S_1^{SSET} be the simulator used to obtain P_1's view in secure string equality test. Simulator S_1 invokes the simulator S_1^{SSET} upon input s' in which s' is the secret reconstructed by P_1. We have that S_1 outputs $(t, m, S_1^{OT}(t, s^\sigma), S_1^{SSET}(s'))$. We now should prove that

$$\{S_1(t, m, S_1^{OT}(t, s^\sigma), S_1^{SSET}(s'))\} \overset{c}{\equiv} \{View_1(t, m, R_1^{OT}(t, s^\sigma), R_1^{SSET}(s'))\}.$$

where $R_1^{OT}(t, s^\sigma)$ denotes the incoming messages of P_1 from the appropriate real oblivious transfer execution, $R_1^{SSET}(s')$ denotes the incoming messages from real string equality test execution.

Observing that the only difference between two distributions above is that the simulator S_1 randomly selects a pattern p instead of using the real input of P_2, so secret sharing share in the input of S_1^{OT} are different from it in the input of R_1^{OT}. However, according to oblivious transfer protocol, the view of P_1 in OT can be generated without knowing the input of P_2, which means that the simulation can be completed without using the secret sharing shares. Assuming that a probabilistic polynomial-time adversary can distinguish the two distributions, it means that it can learn the bits of P_2 which is contrary to the security of oblivious transfer protocol.

Specifically, we first prove the security of the simulated views for OT. As we can see, OT are needed to execute $m(n-m+1)$ times and $S_{1,i}^{OT}$ means the ith execution of 1-out-of-2 OT. We define a hybrid distribution $H_i, i \in \{1, \ldots, m(n-m+1)\}$ in which the first i OTs are simulated and the last $m(n-m+1)-i$ are real. Then, let $H_i(t, m)$ denote the distribution

$$\{t, m, S_{1,1}^{OT}(t, s_1^\sigma), \ldots, S_{1,i}^{OT}(t, s_i^\sigma), R_{1,i+1}^{OT}(t, s_{i+1}^\sigma), \ldots, R_{1,m(n-m+1)}^{OT}(t, s_{m(n-m+1)}^\sigma)\}$$

where $s_i^\sigma, \sigma \in \{0,1\}, i \in \{1, \ldots, m(n-m+1)\}$ denotes secret sharing shares. Notice that $H_{m(n-m+1)}(t, m)$ equals the distribution of $S_1(t, m, S_1^{OT}(t, s^\sigma), S_1^{SSET}(s'))$ and $H_0(t, m)$ is exactly the same as

$$View_1(t, m, R_1^{OT}(t, s^\sigma), R_1^{SSET}(s')).$$

We now prove that $\{H_0(t, m) \overset{c}{\equiv} H_{m(n-m+1)}(t, m)\}$. By contradiction, assume that there exists a PPT distinguisher D and a polynomial $p(\cdot)$ such that,

$$|Pr[D(H_0(t, m)) = 1] - Pr[D(H_{m(n-m+1)}(t, m)) = 1]| > \frac{1}{p(m(n-m+1))}.$$

It follows that there exists an i such that for t, m,

$$|Pr[D(H_i(t,m))=1]-Pr[D(H_{i+1}(t,m))=1]| > \frac{1}{(m(n-m+1))p(m(n-m+1))}.$$

Now, using D to contradict the security of the OT protocol. We note that the only difference between $H_i(t,m)$ and $H_{i+1}(t,m)$ is that message transcript of the $i+1$-th OT are according to $R_1^{OT}(t, s_{i+1}^{\sigma})$ in H_i and according to $S_1(t, s_{i+1}^{\sigma})$ in H_{i+1}. However, we can easily see that for infinitely many inputs, it is possible to distinguish P_1's view in real OT execution from its simulated view with the same probability that it is possible to distinguish $H_i(t,m)$ from $H_{i+1}(t,m)$. It contradicts the security of the OT protocol. We therefore conclude that $\{H_0(t,m) \stackrel{c}{\equiv} H_{m(n-m+1)}(t,m)\}$.

Similarly, according to the characteristics of secure string equality test, the view of P_1 can be generated without knowing the input of P_2. Even if the secret reconstructed by P_1 is corresponding to the random pattern p, the two distributions are also computationally indistinguishable.

P_2 Is Corrupted: In this case, we construct a simulator S_2 that is given inputs of a pattern $p \in \{0,1,*\}^m$, an integer n and output of *result* which is the output of ideal function $\mathcal{F}_{SSET}$ and *result* $\in \{0,1\}$. Then S_2 generates the view of P_2.

Let S_2^{OT} be the simulator that is used for party P_2 in OT protocol. Simulator S_2 invokes the simulator S_2^{OT} upon input p in which p is the pattern. Simulator S_2 performs different operations according to *result*. Specifically, *result* $= 1$ indicates that the substring of text t is successfully matched with the pattern p. At this time, S_2 needs to construct a text t in accordance with p, the text t needs to satisfy that the non-wildcard bits are matched with the pattern and the wildcard bits are randomly selected. *result* $= 0$ indicates that the substring of text t fails to match the pattern p, then simulator S_2 needs to randomly selects a text t as the input of $\mathcal{F}_{OT_2^1}$.

Let S_2^{SSET} be the simulator that is used for party P_2 in secure string equality test. Simulator S_2 invokes the simulator S_2^{SSET} upon input and output $(s, result)$ in which s is the secret randomly selected by P_2 and *result* is the output of $\mathcal{F}_{SSET}$. Similarly, *result* $= 1$ indicates that the substring of text t is successfully matched with the pattern p and S_2 needs to select the same secret s as P_2. *result* $= 0$ indicates that the substring of text t fails to match the pattern p, then simulator S_2 needs to randomly selects a secret. We therefore have that S_2 outputs $(p, n, result, S_2^{OT}(p), S_2^{SSET}(s, result))$. We should prove that the output of simulator S_2 is computationally indistinguishable from the view of P_2, that is

$$\{S_2(p,n,result,S_2^{OT}(p),S_2^{SSET}(s,result))\} \stackrel{c}{\equiv} \{View_2(p,n,result,\mathcal{R}_2^{OT}(p),R_2^{SSET}(s,result))\}.$$

where $R_2^{OT}(p)$ denotes the incoming messages of P_2 from the appropriate real oblivious transfer execution, $R_2^{SSET}(s, result)$ denotes the incoming messages from real string equality test execution.

Observing that S_2 always selects inputs of oblivious transfer protocol and string equality test based on *result*, that is, the *result* between S_2 and $View_2$ is always the same. Taking *result* $= 1$ as an example, it means that only wildcard bits of p are different from t which are constructed by S_2. However, the wildcard bits are irrelevant to the matching result when OT protocol is executed and the s selected by S_2 when *result* $= 1$ is the same as the secret of P_2. Similarly, simulator S_2 randomly selects a text and a secret when *result* $= 0$ which means the matching is unsuccessful. According to the security of oblivious transfer protocol and string equality test, the view of P_2 can be generated without knowing the input of P_1. So the distributions above are also computationally indistinguishable.

Specifically, we also prove a hybrid argument over the simulated views for the OTs. We define a hybrid distribution $H_i', i \in \{1,\ldots,m(n-m+1)\}$ in which the first i OTs are simulated and the last $m(n-m+1)-i$ are real. Formally, let $H_i'(p, n, result)$ denote the distribution

$$\{p, n, result, S_{2,1}^{OT}(p), \ldots, S_{2,i}^{OT}(p), R_{2,i+1}^{OT}(p), \ldots, R_{2,m(n-m+1)}^{OT}(p)\}$$

Notice that $H'_{m(n-m+1)}(p, n, result)$ equals the distribution of $S_2(p, n, result, S_2^{OT}(p), S_2^{SSET}(s, result))$ and $H_0'(p, n, result)$ is exactly the same as

$$View_2(p, n, result, R_2^{OT}(p), R_2^{SSET}(s, result)).$$

We now prove that

$$\{H_0'(p,n,result) \stackrel{c}{\equiv} H'_{m(n-m+1)}(p,n,result)\}.$$

By contradiction, assume that there exists a PPT distinguisher D' and a polynomial $p'(\cdot)$ such that,

$$|Pr[D'(H_0'(p,n,result))=1] - Pr[D'(H'_{m(n-m+1)}(p,n,result))=1]| > \frac{1}{p'(m(n-m+1))}.$$

It follows that there exists an i such that for $p, n, result$,

$$|Pr[D'(H_i'(p,n,result))=1]-Pr[D'(H'_{i+1}(p,n,result))=1]| > \frac{1}{(m(n-m+1))p'(m(n-m+1))}.$$

We now use D' to contradict the security of the oblivious transfer protocol. First, note that the only difference between $H_i'(p, n, result)$ and $H'_{i+1}(p, n, result)$ is that message transcript of the $i+1$-th OT are according to $R_2^{OT}(p)$ in H_i' and according to $S_2(p)$ in H'_{i+1}. However, we can easily see that for infinitely many inputs, it is possible to distinguish the view of P_2 in a real OT execution from its simulated view with the same probability that it is possible to distinguish $H_i'(p, n, result)$ from $H'_{i+1}(p, n, result)$. It contradicts the security of the OT protocol. We therefore conclude that $\{H_0'(p, n, result) \stackrel{c}{\equiv} H'_{m(n-m+1)}(p, n, result)\}$.

TABLE 1. Efficiency comparisons.

Protocol	tools	Computation	Communication
Hazay [20]	ElGmal PKE	$O(nm)$	$O(nm)$
Baron [22]	Additively HE	$O(nm)$	$O(n\tau)$
Kolesnikv [25]	OT extension	$O(k)$	$O(nm)$
Ours	OT extension	$O(k)$	$O(nm)$

Similarly, according to the characteristics of secure string equality test, the view of P_2 can be generated without knowing the input of P_1. So the distributions above are also computationally indistinguishable.

Now we complete the formal proof of theorem 1. □

V. EFFICIENCY AND EXPERIMENT

A. EFFICIENCY AND COMPARISON

In our protocol, it requires a total of $O(nm)$ OT operations (the secure string equality test can be implemented with OT [32]). However, if we use OT extension technique, the number of OTs can be greatly reduced to $O(k)$ level where k is security parameter, essentially independent of the number of OTs and can be as small as 80 or 128 [33]. In addition, the communication complexity of this protocol is also $O(nm)$.

In Table 1, we show the comparison between our protocol and previous wildcard pattern matching protocol. Firstly, we consider cryptographic tools. Reference [14] is based on distributed ElGamal encryption scheme and it needs to implement a joint decryption protocol, [17] is based on homomorphic encryption scheme, [20] is based on OT extension protocol and needs to invoke secure string equality test. In terms of computational complexity, [14] and [17] requires $O(nm)$ operations, where n and m are input length of two participants. The computational complexity of [20] and our protocol is $O(k)$, where $k = 128$ is security parameter and is smaller than nm, which benefits from OT extension technology. In addition, the communication complexity of [17] is $O(n\tau)$ where τ is statistical security parameter and in the range of about 1024-2048. The communication complexity of [14] and [20] and our protocol are $O(nm)$. Through comparison, we can see that our protocol has the same efficiency as that in [20]. However, in our protocol, the offline phase only requires XOR operations on bit strings without defining other data structures, so it is very easy to implement by hardware and more suitable for low-cost IoT devices.

B. EXPERIMENTS

In this section, we implemented our protocol in personal computer. The computer is equipped with an Intel Core i7- 6700 processor and 16GBs of RAM. The programming language is C++ based on 64-bit architecture. In offline phase, P_2 runs secret splitting algorithm of XOR-secret-sharing scheme, spliting a λ-bit binary string into m shares. P_1 runs secret reconstruction algorithm, combining m shares into a secret. In the whole offline phase, only XOR operations of bit strings are executed, so it is very efficient. Experiment results in this phase are shown in Fig.5. In online phase, it mainly includes oblivious transfer and secure string equality test. Because secure string equality test can be implemented by oblivious transfer, we only test the efficiency of oblivious transfer. In this phase, we used OT extension technology instead of a large number of parallel execution OT protocols [34]. The results are shown in Fig.6. As shown in the two figures, the x-coordinate represents the size of P_2's input m (i.e. the number of bits of m), and the y-coordinate represents the corresponding execution time.

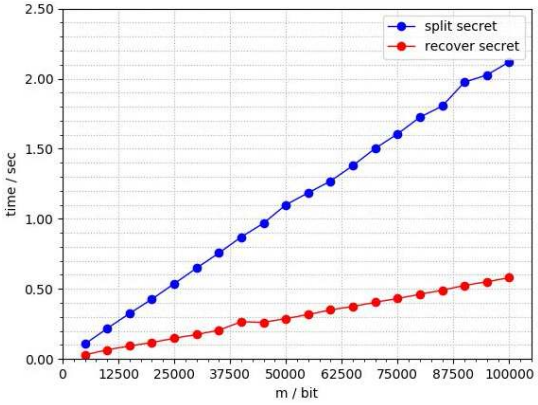

FIGURE 5. Execution time of secret sharing.

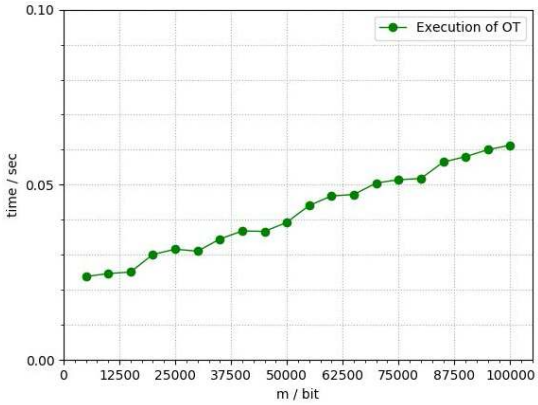

FIGURE 6. Execution time of string OT with OT extension.

VI. APPLICATION IN SMART MEDICAL

In recent years, the imperfect medical management system, high medical costs and polarized medical resources have brought many social problems. These problems have become an important factor affecting the harmonious development of society. We urgently need to establish a smart medical platform, so that patients can use shorter waiting time and pay for basic medical expenses to enjoy safe, convenient and high-quality medical services. As a development direction of smart medical, wearable devices can conduct health

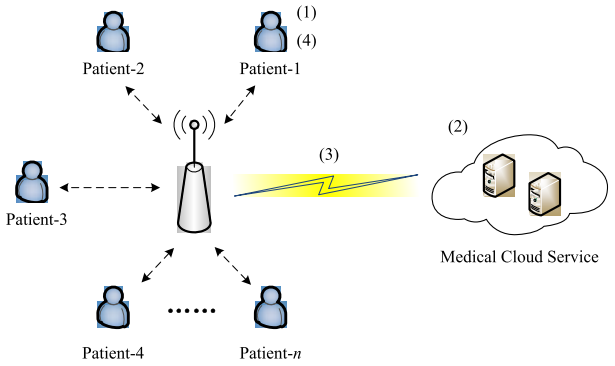

FIGURE 7. Application in smart medical.

management and disease information prediction. Specifically, the wearable device collects a large amount of health data and interacts with healthcare cloud service to analyze these data. Pattern matching is a common method for processing data. Through pattern matching, we can judge whether the data collected by wearable device conforms to certain characteristics. If there exists successful matching, sever can return health reports or warnings. This process is as follows and shown in Fig.7.

1. The patient wears wearable device, which collects patient's health data all the time. For the execution of π_{WPM}, device encodes the data into binary string (called pattern) and the string is exactly device's input to this protocol.
2. The medical cloud service holds mass data about health characteristics, and encodes these data into binary text. The text acts as the input of π_{WPM} of cloud service.
3. Wearable device interacts with medical cloud service by wireless transmission. Then, device and cloud service implement our π_{WPM} protocol jointly and return the matching result to wearable device.
4. According to the result, patient can obtain their health condition.

VII. CONCLUSION

This paper mainly considers the construction of secure wildcard pattern matching protocol in semi-honest adversary model. Our protocol is based on three cryptographic tools which are secret sharing, oblivious transfer and secure string equality test. Through the setting of secret sharing shares, the wildcard bits and non-wildcard bits are represented respectively. Combined with OT protocol, the wildcard pattern matching is converted into exact pattern matching, and finally the string equality test is called to determine whether the matching is successful. Due to the use of OT extension technology, all offline operations in our protocol are bit operations, which are very efficient and suitable for lightweight IoT devices.

REFERENCES

[1] P. Cerwall and P. Jonsson. (2015). *Ericsson Mobility Report*. [Online]. Available: http://www.ericsson.com/res/docs/2015/mobility-report/ericsson-mobility-report-nov-2015.pdf

[2] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 122–129, Jan. 2017. doi: 10.1109/MCOM.2017.1600267CM.

[3] K. Su, J. Li, and H. Fu, "Smart city and the applications," in *Proc. Int. Conf. Electron., Commun. Control (ICECC)*, Sep. 2011, pp. 1028–1031.

[4] X. Li, J. Niu, S. Kumari, F. Wu, and K.-K. R. Choo, "A robust biometrics based three-factor authentication scheme for global mobility networks in smart city," *Future Gener. Comput. Syst.*, vol. 83, pp. 607–618, Jun. 2018.

[5] P. Kumar and H.-J. Lee, "Security issues in healthcare applications using wireless medical sensor networks: A survey," *Sensors*, vol. 12, no. 1, pp. 55–91, 2012.

[6] M. Haghi, K. Thurow, and R. Stoll, "Wearable devices in medical Internet of Things: Scientific research and commercially available devices," *Healthcare Inf. Res.*, vol. 23, no. 1, pp. 4–15, 2017.

[7] X. Li, F. Wu, M. K. Khan, L. Xu, J. Shen, and M. Jo, "A secure chaotic map-based remote authentication scheme for telecare medicine information systems," *Future Gener. Comp. Syst.*, vol. 84, pp. 149–159, Jul. 2018. doi: 10.1016/j.future.2017.08.029.

[8] X. Li, M. H. Ibrahim, S. Kumari, A. K. Sangaiah, V. Gupta, and K.-K. R. Choo, "Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks," *Comput. Netw.*, vol. 129, pp. 429–443, Dec. 2017.

[9] X. Li, J. Peng, J. Niu, F. Wu, J. Liao, and K. R. Choo, "A robust and energy efficient authentication protocol for industrial Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1606–1615, Jun. 2018. doi: 10.1109/JIOT.2017.2787800.

[10] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ECC-based provable secure authentication protocol with privacy preserving for industrial Internet of Things," *IEEE Trans. Ind. Inform.*, vol. 14, no. 8, pp. 3599–3609, Aug. 2018. doi: 10.1109/TII.2017.2773666.

[11] J. R. Troncoso-Pastoriza, S. Katzenbeisser, and M. U. Celik, "Privacy preserving error resilient DNA searching through oblivious automata," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, Alexandria, Virginia, USA, Oct. 2007, pp. 519–528. doi: 10.1145/1315245.1315309.

[12] R. Gennaro, C. Hazay, and J. S. Sorensen, "Text search protocols with simulation based security," in *Proc. Int. Workshop Public Key Cryptogr.* Paris, France, May 2010, pp. 332–350. doi: 10.1007/978-3-642-13013-7_20.

[13] P. Mohassel, S. Niksefat, S. S. Sadeghian, and B. Sadeghiyan, "An efficient protocol for oblivious DFA evaluation and applications," in *Proc. Cryptographers Track RSA Conf.*, San Francisco, CA, USA, Feb./Mar. 2012, pp. 398–415. doi: 10.1007/978-3-642-27954-6_25.

[14] L. Wei and M. K. Reiter, "Third-party private DFA evaluation on encrypted files in the cloud," in *Proc. Eur. Symp. Res. Comput. Secur.*, Pisa, Italy, Sep. 2012, pp. 523–540. doi: 10.1007/978-3-642-33167-1_30.

[15] H. Sasakawa, H. Harada, D. duVerle, H. Arimura, K. Tsuda, and J. Sakuma, "Oblivious evaluation of non-deterministic finite automata with application to privacy-preserving virus genome detection," in *Proc. 13th Workshop Privacy Electron. Soc.*, Scottsdale, AZ, USA, Nov. 2014, pp. 21–30. doi: 10.1145/2665943.2665954.

[16] F. Chen *et al.*, "Secure hashing-based verifiable pattern matching," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2677–2690, Nov. 2018. doi: 10.1109/TIFS.2018.2825141.

[17] C. Hazay and Y. Lindell, "Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries," in *Proc. Theory Cryptogr. Conf.*, New York, USA, Mar. 2008, pp. 155–175. doi: 10.1007/978-3-540-78524-8_10.

[18] C. Hazay and Y. Lindell, "Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries," *J. Cryptol.*, vol. 23, no. 3, pp. 422–456, Jul. 2010. doi: 10.1007/s00145-008-9034-x.

[19] J. Katz and L. Malka, "Secure text processing with applications to private DNA matching," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, Chicago, Il, USA, Oct. 2010, pp. 485–492. doi: 10.1145/1866307.1866361.

[20] C. Hazay and T. Toft, "Computationally secure pattern matching in the presence of malicious adversaries," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Singapore, Dec. 2010, pp. 195–212. doi: 10.1007/978-3-642-17373-8_12.

[21] C. Hazay and T. Toft, "Computationally secure pattern matching in the presence of malicious adversaries," *J. Cryptol.*, vol. 27, no. 2, pp. 358–395, Apr. 2014. doi: 10.1007/s00145-013-9147-8.

[22] J. Baron, K. E. Defrawy, K. Minkovich, R. Ostrovsky, and E. Tressler, "5pm: Secure pattern matching," in *Proc. Int. Conf. Secur. Cryptogr. Netw.*, 2012, pp. 222–240. doi: 10.1007/978-3-642-32928-9_13.
[23] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Secure pattern matching using somewhat homomorphic encryption," in *Proc. ACM Workshop Cloud Comput. Secur. Workshop*, Berlin, Germany, Nov. 2013, pp. 65–76. doi: 10.1145/2517488.2517497.
[24] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Privacy-preserving wildcards pattern matching using symmetric somewhat homomorphic encryption," in *Proc. Australas. Conf. Inf. Secur. Privacy*, Wollongong, NSW, Australia, Jul. 2014, pp. 338–353. doi: 10.1007/978-3-319-08344-5_22.
[25] V. Kolesnikov, M. Rosulek, and N. Trieu. (2017). *SWiM: Secure Wildcard Pattern Matching From OT Extension*. [Online]. Available: http://eprint.iacr.org/2017/1150
[26] J. Darivandpour and M. J. Atallah, "Efficient and secure pattern matching with wildcards using lightweight cryptography," *Comput. Secur.*, vol. 77, pp. 666–674, Aug. 2018.
[27] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Aug. 1979. doi: 10.1145/359168.359176.
[28] G. Blakley, "Safeguarding cryptographic keys," in *Proc. 1979 AFIPS Nat. Comput. Conf.* Montvale, NJ, USA: AFIPS Press, 1979, pp. 313–317.
[29] M. O. Rabin. (2005). *How to Exchange Secrets With Oblivious Transfer*. [Online]. Available: http://eprint.iacr.org/2005/187
[30] R. Fagin, M. Naor, and P. Winkler, "Comparing information without leaking it," *Commun. ACM*, vol. 39, no. 5, pp. 77–85, May 1996. doi: 10.1145/229459.229469.
[31] O. Goldreich. (1998). Secure multi-party computation (manuscript). [Online]. Available: http://www.wisdom.weizmann.ac.il/~oded/pp.html
[32] B. Pinkas, T. Schneider, and M. Zohner, "Faster private set intersection based on OT extension," in *Proc. 23rd USENIX Secur. Symp.*, San Diego, CA, USA, Aug. 2014, pp. 797–812. [Online]. Available: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/pinkas
[33] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank, "Extending oblivious transfers efficiently," in *Proc. Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, Aug. 2003, pp. 145–161. doi: 10.1007/978-3-540-45146-4_9.
[34] X. Wang, A. J. Malozemoff, and J. Katz. (2016). *EMP-Toolkit: Efficient MultiParty Computation Toolkit*. [Online]. Available: https://github.com/emp-toolkit

HONG QIN received the B.S. degree in computer science from Shandong Normal University, China, in 2018, where she is currently pursuing the degree in computer science with the School of Information Science and Engineering. Her primary interests include secure multi-party computation and privacy-preserving machine learning.

HAO WANG received the Ph.D. degree in computer science from Shandong University, China, in 2012. He is currently an Associate Professor with Shandong Normal University. His primary interest includes public key cryptography, in particular, designing cryptographic primitives and provable security. His current research interests include attribute-based cryptography, security in cloud computing, and secure multi-party computation.

XIAOCHAO WEI received the Ph.D. degree in computer science from Shandong University, China, in 2017. He is currently a Lecturer with Shandong Normal University. His main interests include secure multiparty computation, privacy preserving, and searchable encryption.

LIKUN XUE received the B.S. degree in computer science from Jinan University, China, in 2018. He is currently pursuing the degree in computer science with the School of Information Science and Engineering, Shandong Normal University, China. His primary interests include secure multi-party computation and cloud security.

LEI WU received the Ph.D. degree in computer science from Shandong University, China, in 2009. He is currently an Associate Professor with Shandong Normal University. His primary interest is public key cryptography, in particular, designing cryptographic primitives and provable security, and main interests are cloud security and public key cryptography.