A Comment on Privacy-Preserving Scalar Product Protocols as Proposed in "SPOC"

Thomas Schneider and Amos Treiber

Abstract—Privacy-preserving scalar product (PPSP) protocols are an important building block for secure computation tasks in various applications. Lu et al. (TPDS'13) introduced a PPSP protocol that does not rely on cryptographic assumptions and that is used in a wide range of publications to date. In this comment paper, we show that Lu et al.'s protocol is insecure and should not be used. We describe specific attacks against it and, using impossibility results of Impagliazzo and Rudich (STOC'89), show that it is inherently insecure and cannot be fixed without relying on at least some cryptographic assumptions.

Index Terms—Privacy-preserving scalar product protocols, secure computation, oblivious transfer

1 Introduction

The scalar product is a fundamental operation in linear algebra that is used in a variety of fields, e.g., serving as the basis of deep neural networks, biometric characterization, or computer graphics. Suppose two parties P_0 and P_1 with respective input vectors $\vec{a}$ and $\vec{b}$ want to securely compute the scalar product $\vec{a} \cdot \vec{b} = \sum_{i=1}^{n} a_i b_i$ such that P_0 obtains the result $\vec{a} \cdot \vec{b}$ without revealing anything else about $\vec{b}$ to P_0 or anything about $\vec{a}$ to P_1. This *secure two-party computation* of the scalar product is an important building block for preserving privacy in many applications. In 2013, Lu et al. [1] proposed a privacy-preserving scalar product (PPSP) protocol in their paper titled "SPOC: A Secure and Privacy-Preserving Opportunistic Computing Framework for Mobile-Healthcare Emergency". This protocol relies on "multi-party random masking and polynomial aggregation techniques" [2], where absolutely no public-key cryptography is used. In fact, their protocol does not make any cryptographic assumptions at all and the authors claim that it achieves information-theoretic security. As shown in [2], the protocol is much faster than public-key based protocols using homomorphic encryption. Since then, this protocol has been and is still used in many privacy-preserving solutions, e.g., [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], including support vector machines [17], facial expression classification [9], medical prediagnosis [18], and speaker verification [10], [11].

In this comment paper, we present devastating attacks against the original [1] and subsequent [2] versions of Lu et al.'s protocol. Our attacks fully break privacy and show that the protocol should not be used in applications. Before presenting our concrete attacks in Section 3, we first show in Section 2 why Lu et al.'s protocol is *inherently* insecure and can only be fixed if at least some public-key cryptography is used.

2 Lu et al.'s Protocol Cannot Be Secure

A fundamental issue with privacy-preserving tasks is that the absence of attacks does not guarantee privacy. To assure the privacy of new protocols, a formal proof of security is needed. Using established simulation-based security notions, such proofs show that *only* what can be computed from a priori information can be learned by executing the protocol. In this section, we will show that Lu et al.'s protocol cannot be secure under the established security notions.

2.1 Formalizing Secure Two-Party Computation

Formally, the secure two-party computation (STPC) of a function $f(a,b)$ on inputs a from P_0 and b from P_1 by a protocol Π is defined by a *simulator* $S = (S_1, S_2)$ that *simulates* the *views* of the parties participating in Π [21, chapter 7]:

$$\{S_0(a, f(a,b))\}_{a,b} \overset{c}{\approx} \{\text{view}_0^{\Pi}(a,b)\}_{a,b} \text{ and}$$
$$\{S_1(b, f(a,b))\}_{a,b} \overset{c}{\approx} \{\text{view}_1^{\Pi}(a,b)\}_{a,b},$$

where $\overset{c}{\approx}$ denotes *computational indistinguishability*. $S_{i\in\{0,1\}}$ is computationally (ploynomial-time) bounded and needs to simulate view_i^{Π}, which contains all incoming messages received by P_i during the execution of Π. If such a simulator exists, then the protocol is considered secure because everything that can be learned from participating in the protocol ($\{\text{view}_i^{\Pi}(a,b)\}_{a,b}$) can also be learned by information that is known to the party anyway (S_i sees either the input a or b and the output $f(a,b)$). Conversely, if no such simulator exists, then the distribution generated by *any* S can be distinguished from the distribution of the views of the protocol execution, meaning that a party observing the view reveals more information than just knowing its own input and the output. This is the established notion and the de facto standard to model secure computation tasks for privacy-preserving solutions. Thus, in order to ensure the security of a protocol, a security proof of indistinguishability is needed [22]. The model we are concerned with here is in the context of *semi-honest* (or *passive*) security, where P_0 and P_1 honestly follow the protocol but try to learn additional information.

In the above definition, it suffices to show that one party can distinguish between different inputs of the other party based on the observed execution of a protocol to break its privacy. For instance, in our specific attacks against Lu et al.'s privacy-preserving protocol (cf. Section 3.2), we will show that P_0 can distinguish between different inputs of P_1 regardless of the output, thereby learning more than the minimal amount of information implied by the input and the output. Because this additional information is hard to specify and highly depends on the use case, protocols where this distinction is possible are considered insecure.

2.2 A secure PPSP Protocol has to Rely on Cryptographic Hardness Assumptions

In the following, we will put Lu et al.'s PPSP protocol in relation to well-established cryptographic primitives, showing that PPSP has to rely on public-key cryptography. A summary of these relations can be found in Fig. 1.

More precisely, PPSP is closely related to a primitive called oblivious transfer (OT). In OT, a party P_0 inputs a choice bit b and P_1 inputs two bits (x_0, x_1). P_0 receives x_b as output without learning any information about x_{1-b} and without revealing any information about b to P_1. OT is a strong primitive that implies many more fundamental cryptographic building blocks such as STPC [24].

Of course, STPC can be used to realize PPSP and known and *secure* PPSP protocols usually rely on STPC based on homomorphic encryption or OT [25]. Since OT implies STPC, it follows that OT implies PPSP. Conversely, the existence of a PPSP protocol would imply OT, as OT is just a special case of PPSP where $\vec{a} = (\bar{b}, b)$ and $\vec{b} = (x_0, x_1)$. Therefore, PPSP is equivalent to OT and requires the

• *The authors are with the Cryptography and Privacy Engineering Group (ENCRYPTO), TU Darmstadt, Darmstadt 64289, Germany.*
E-mail: {schneider, treiber}@encrypto.cs.tu-darmstadt.de.

Manuscript received 14 Dec. 2018; revised 24 Aug. 2019; accepted 31 Aug. 2019. Date of publication 3 Sept. 2019; date of current version 10 Jan. 2020.
(Corresponding author: Amos Treiber.)
Recommended for acceptance by M. Parashar.
Digital Object Identifier no. 10.1109/TPDS.2019.2939313

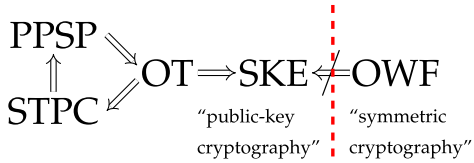

Fig. 1. Relations of privacy-preserving scalar product (PPSP), oblivious transfer (OT), secure two-party computation (STPC), symmetric key exchange (SKE), and one-way functions (OWF). The black box separation between "public-key" and "symmetric cryptography" shows that SKE cannot be based on OWF [23]. Therefore, a PPSP protocol has to be based on public-key cryptographic assumptions.

same assumptions required for OT like, e.g., public-key cryptography, noisy channels, or hardware tokens.

OT can also be used to implement symmetric key exchange (SKE) [26], [27]. Impagliazzo and Rudich [23] proved that a black-box reduction of SKE to one-way functions (the central building block of symmetric cryptography) would imply $P \neq NP$. This means that SKE and thus OT *very likely* require at least some complexity-theoretic assumptions of public-key cryptography, as otherwise a proof of $P \neq NP$ would be found. As such, all PPSP protocols that rely solely on symmetric cryptography or make no cryptographic hardness assumptions at all (like Lu et al.'s protocol) must be flawed.

3 Lu et al.'s Protocol Is Insecure

Lu et al.'s PPSP protocol first appeared in [1] as a sub-protocol in a privacy-preserving healthcare framework and was later extended in [2] by introducing fixes to preserve privacy. The protocol is shown in Fig. 2, with the extensions of [2] underlined. Before presenting our specific attacks, we briefly outline how the protocol works.

3.1 How the Protocol is Supposed to Work

Correctness stems from the observation that $E = \sum_{a_i \neq 0, b_i \neq 0} a_i b_i \alpha^2 + \sum_{a_i=0, b_i \neq 0} b_i c_i \alpha + \sum_{a_i \neq 0, b_i = 0} r_i (a_i \alpha + c_i) + \sum_{a_i = 0, b_i = 0} r_i c_i$ and therefore $E \bmod \alpha^2$ contains all addends that are not multiples of α^2, i.e., all addends except $\sum_{a_i \neq 0, b_i \neq 0} a_i b_i \alpha^2$. Thus, $\vec{a} \cdot \vec{b} = \frac{E - (E \bmod \alpha^2)}{\alpha^2}$ under the constraint that $\sum_{a_i \neq 0, b_i \neq 0} a_i b_i \alpha^2 + \sum_{a_i=0, b_i \neq 0} b_i c_i \alpha + \sum_{a_i \neq 0, b_i = 0} r_i (a_i \alpha + c_i) + \sum_{a_i = 0, b_i = 0} r_i c_i < p$ and $\sum_{a_i=0, b_i \neq 0} b_i c_i \alpha + \sum_{a_i \neq 0, b_i = 0} r_i (a_i \alpha + c_i) + \sum_{a_i = 0, b_i = 0} r_i c_i < \alpha^2$. To make the analysis of our attacks easier, we translate the latter inequality onto the corresponding bit-length parameters, resulting in the following conditions *necessary* for correctness:

$$\log_2 n + \log_2 q + k_3 < k_2, \tag{1a}$$

$$\log_2 n + \log_2 q + k_4 < k_2, \tag{1b}$$

$$\log_2 n + k_3 + k_4 < 2k_2. \tag{1c}$$

A violation of any of the above inequalities would result in the protocol being incorrect for some or even all inputs. The parameters used for randomly masking the inputs, k_3 and k_4, are both set to 128 to allow for a randomness source of 128-bit. As a result of the above constraints when assuming an input space of $n = q = 2^{32}$, the parameters are set to $k_1 = 512$ and $k_2 = 200$ in [2] to ensure correctness. Similar assumptions can be found in the original protocol [1].

The protocol's security is entirely based on masking values with random addends or factors. In the first step, P_0 masks all values C_i by multiplying with s. For $a_i = 0$, just a random c_i is masked, while a random c_i added to $\alpha \cdot a_i$ is masked otherwise. The intention behind s and all c_i is to hide any information about a_i and, indeed, it is impossible to distinguish between different a_i based on the uniform distribution from which all c_i are drawn. α and p serve no security purpose but ensure correctness. In step 2, P_1 either randomizes C_i by multiplying with the random r_i or it just multiplies $b_i \cdot \alpha$ to C_i. The supposed idea here is that, because $b_{n+1} = b_{n+2} = 0$, $\sum_i C_i$ is randomized by the addends $r_{n+1} \cdot C_{n+1}$ and $r_{n+2} \cdot C_{n+2}$. Thus it *seems* that different values of D from different $\vec{b}$ should not be distinguishable. There exist some proof sketches of the protocols in [1], [2] and some of the works building on them. The security analyses do not rely on the established indistinguishability-based security notions presented in Section 2.1, but instead make use of ad-hoc security notions that are based around the principle that the input cannot be reconstructed. Below, we will present specific distinguishing attacks that even allow P_0 to *check* whether P_1's input is a candidate $\vec{b}$. This obviously violates privacy and shows that contrary to the established primitives, the ad-hoc security definitions used for the proofs do not capture any useful sense of privacy.

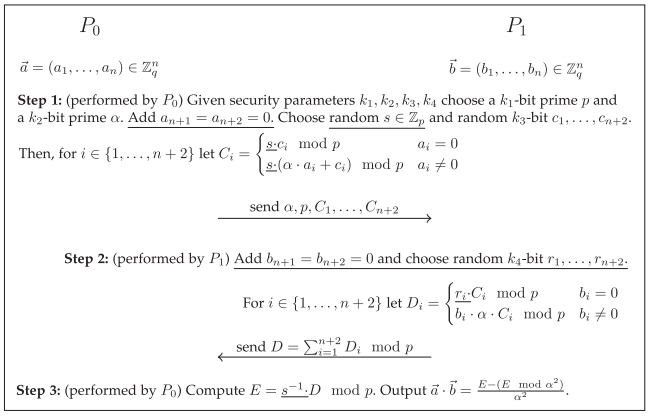

Fig. 2. Lu et al.'s PPSP protocol [1] with the extensions of [2] (underlined).

3.2 Our Specific Attacks

One can immediately see why the original protocol of [1] is broken: $D = \sum_{b_i=0} C_i + \sum_{b_i \neq 0} b_i \cdot \alpha \cdot C_i$. Since D is completely deterministic and depends only on α, C_i, and $\vec{b}$, party P_0 can easily distinguish different values of $\vec{b}$ because it knows α and all C_i. For instance, for $\vec{b} = \vec{0}$, P_1 will return $\sum_{i=1}^{n} C_i$ whereas for $\vec{b'} = (1, 0, \ldots, 0)$, P_1 will return $\alpha \cdot C_1 + \sum_{i=2}^{n} C_i$. This attack works for any value of $\vec{a}$.

3.2.1 Attack on the Fixed Protocol for $\vec{a} = \vec{0}$

The above vulnerability was fixed in [2] by introducing random addends to D via $b_{n+1} = b_{n+2} = 0$. Operations based on public-key cryptography still do not appear in the protocol. Thus, the security of this version is implausible as well (cf. Section 2.2). Indeed, we found another attack that can distinguish different $\vec{b}$. At first, we consider this attack for the case of $\vec{a} = \vec{0}$, because then the output of the ideal functionality is equal to 0 and yields no knowledge about $\vec{b}$. In that case, the ability to distinguish any distinct $\vec{b}$ clearly demonstrates that information about the inputs is leaked. Using the following strategy, P_0 can distinguish between $\vec{b} = \vec{0}$ and $\vec{b'} = (1, 0, \ldots, 0)$ after computing E in step 3:

$$\text{If } E/\alpha \approx c_1 \text{then output guess } \vec{b'}, \text{ else output guess } \vec{b}.$$

Our attack relies on the different sizes of the parameters and works because they create a conflict between security and correctness: to prevent our attack, the parameters have to be changed in such a way that they violate the correctness constraints. More specifically, for $\vec{b'}$, P_0 receives $E = \alpha \cdot c_1 + \sum_{i=2}^{n+2} r_i c_i$. In our attack, P_0 will compute $E/\alpha = c_1 + \frac{\sum_i r_i c_i}{\alpha}$. Since $|c_1| = k_3$ and $|\frac{\sum_i r_i c_i}{\alpha}| = \log_2 n + k_3 + k_4 - k_2$, $E/\alpha \approx c_1$ except for the $\log_2 n + k_3 + k_4 - k_2$ least significant bits. Conversely, for $\vec{b}$, P_0 receives $E = \sum_{i=1}^{n+2} r_i c_i$ and thus will only obtain some $|E/\alpha| = \log_2 n + k_3 + k_4 - k_2$ bit integer. Hence, to make our distinction impossible, the parameters need to

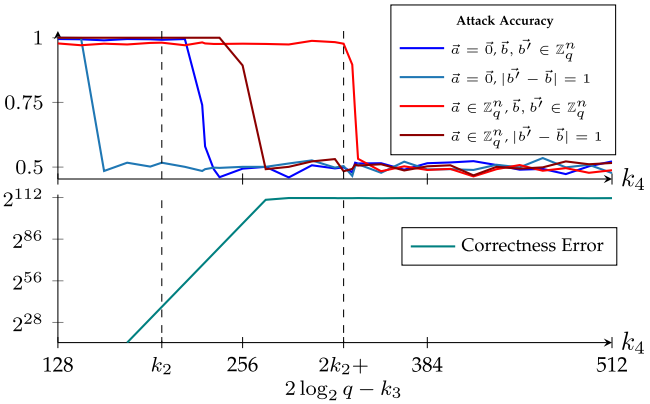

Fig. 3. Correctness of Lu et al.'s protocol [2] (absolute error) for $\vec{b} = \vec{0}$ and accuracy of our attacks in distinguishing two distinct $\vec{b}, \vec{b'}$, given for $n = 256$, $q = 2^{32}$, $k_1 = 512, k_2 = 200, k_3 = 128$, and varying k_4. Vectors are created uniformly at random, unless indicated otherwise. As predicted, the accuracies of our attacks against random $\vec{b}, \vec{b'}$ drop after $k_2 \leq k_4$ and $2(k_2 + \log_2 q) - k_3 \leq k_4$, but at this point the protocol already produces incorrect results.

satisfy *at least* $k_3 \leq \log_2 n + k_3 + k_4 - k_2 \Leftrightarrow k_2 - \log_2 n \leq k_4$, which violates Equation 1b necessary for correctness.

The attack can also be extended to distinguish between *any* $\vec{b}$ by checking whether $E/\alpha \approx \sum_i b_i c_i$. Similar to the reasoning above, the attack can only be prevented if at least $k_2 \leq k_4$, which also violates Equation 1b. Not only does this break privacy because it allows for distinguishing any $\vec{b}$, this also enables an adversary to check whether a suspected input $\vec{b}$ is the real one.

3.2.2 Attack on the Fixed Protocol for any $\vec{a}$

Even though the previous attack is enough to violate privacy, we will further show how to adapt it when using any $\vec{a}$ as input. Knowing its own input $\vec{a}$ and the suspected input $\vec{b}$, P_0 just checks whether $E/\alpha \approx \sum_{a_i \neq 0} a_i b_i \alpha + \sum_{a_i = 0} b_i c_i$. Analogously to the analysis in Section 3.2.1, this distinction could only fail if $|\sum_{a_i \neq 0} a_i b_i \alpha + \sum_{a_i = 0} b_i c_i| \leq |\sum_{a_i \neq 0, b_i = 0} r_i c_i + \sum_{a_i = 0, b_i = 0} \frac{r_i c_i}{\alpha}|$ which, taking into account Equations 1a and 1b, requires that $2(k_2 + \log_2 q) - k_3 \leq k_4$. This would contradict Equation 1c and therefore violate correctness.

3.2.3 Evaluation

To show the feasibility of our attacks, we implemented them alongside the protocol. Our implementation shows that for the parameters used in [2], any user input $\vec{b}$ can easily be distinguished and even detected by P_0. The implementation is freely available as open source and can be found online at https://encrypto.de/code/SPOCattack.

We also evaluate the protocol's correctness as well as the effectiveness of our attacks depending on varying parameters. The results are presented in Fig. 3 for varying values of k_4 and confirm the contradiction between the protocol's correctness and its security. Under the correctness constraints of Equations 1a, 1b, 1c, all of our attacks are close to 100 percent accurate. Conversely, the protocol is entirely correct for these parameter choices as well. When Equation 1b is broken with $k_4 \geq k_2 - \log_2 q - \log_2 n$, the absolute correctness error starts to appear and rises rapidly. Shortly after this, when the accuracy threshold $k_4 \geq k_2$ of our first attack distinguishing any $\vec{b}$ is passed, its accuracy quickly drops to the baseline of 50 percent (the accuracy of randomly guessing between two $\vec{b}$). The same occurs after the threshold for our second distinguishing attack is passed, at which point the maximum correctness error of $k_1 - 2k_2 = 112$ bit is already reached. Furthermore, to demonstrate that our attacks even allow to test for a certain $\vec{b}$, we also evaluate both attacks by distinguishing a random $\vec{b}$ from a $\vec{b'}$ that only differs from $\vec{b}$ by 1 in one position. Our evaluation shows that, though the accuracies drop earlier than for random $\vec{b'}$, these attacks work for the standard parameters and that therefore a precise testing and searching for P_1's input is possible. Notably, we evaluate the correctness for $\vec{b} = \vec{0}$, as Equation 1b comes from the random addends resulting from all $b_i = 0$. When using a completely random $\vec{b}$, the correctness error only starts to appear at $k_4 \geq k_2 + 64 = 264$ but, since the protocol should be correct for any input, we display the results for $\vec{b} = \vec{0}$.

Our implementation establishes that in any application using the protocol, P_0 can check whether P_1 has a certain input (like, e.g., a certain illness in a healthcare application). Clearly, this is a severe violation of privacy and serves as a reminder that the security notions used by the protocol's security analysis (cf. Section 3.1) are insufficient and that the established definitions based on indistinguishability (cf. Section 2.1) should be used instead. As outlined in Section 2.2, similar attacks will inadvertently still be possible even if additional randomizations are introduced to prevent these concrete attacks as long as no cryptographic assumptions are utilized.

4 Conclusion

We showed in Section 2.2 that protocols for the secure two-party computation of the scalar product imply oblivious transfer. As a result, such protocols very likely require public-key cryptography. Lu et al.'s protocol [1], [2] is an example in academic use today that does not rely on such assumptions and is thus inherently insecure. Indeed, we found specific attacks that we have verified with an implementation, showing that their protocol does not guarantee privacy. With this comment paper we want to stress that (at least some) expensive public-key cryptography is necessary for such protocols and that new protocols should be proven secure in established formal frameworks to catch such flaws.

Acknowledgments

We want to thank the anonymous reviewers for their helpful feedback. This work was supported by the DFG as part of project A.1 within the RTG 2050 "Privacy and Trust for Mobile Users" and as part of project E4 within the CRC 1119 CROSSING, and by the BMBF and the HMWK within CRISP.

References

[1] R. Lu, X. Lin, and X. Shen, "SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 3, pp. 614–624, Mar. 2013.

[2] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in big data era," *IEEE Netw.*, vol. 28, no. 4, pp. 46–50, Jul./Aug. 2014.

[3] C. Huang, R. Lu, H. Zhu, J. Shao, and X. Lin, "FSSR: Fine-grained EHRs sharing via similarity-based recommendation in cloud-assisted eHealthcare system," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2016, pp. 95–106.

[4] H. Kaur, N. Kumar, and S. Batra, "ClaMPP: A cloud-based multi-party privacy preserving classification scheme for distributed applications," *J. Supercomputing*, vol. 75, no. 6, pp. 3046–3075, 2019.

[5] X. Liu, H. Zhu, R. Lu, and H. Li, "Efficient privacy-preserving online medical primary diagnosis scheme on naive Bayesian classification," *Peer-to-Peer Netw. Appl.*, vol. 11, no. 2, pp. 334–347, 2018.

[6] T. H. Luan, R. Lu, X. Shen, and F. Bai, "Social on the road: Enabling secure and efficient social networking on highways," *IEEE Wireless Commun.*, vol. 22, no. 1, pp. 44–51, Feb. 2015.

[7] E. Luo, Q. Liu, and G. Wang, "NMHP: A privacy preserving profile matching protocol in multi-hop proximity mobile social networks," in *Proc. Int. Conf. Algorithms Archit. Parallel Process.*, 2015, pp. 463–474.

[8] Y. Rahulamathavan and M. Rajarajan, "Hide-and-seek: Face recognition in private," in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 7102–7107.

[9] Y. Rahulamathavan and M. Rajarajan, "Efficient privacy-preserving facial expression classification," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 3, pp. 326–338, May/Jun. 2017.

[10] S. Rahulamathavan, X. Yao, R. Yogachandran, K. Cumanan, and M. Rajarajan, "Redesign of Gaussian mixture model for efficient and privacy-preserving speaker recognition," in *Proc. Int. Conf. Cyber Situational Awareness Data Analytics Assessment*, 2018, pp. 1–8.

[11] Y. Rahulamathavan, K. Sutharsini, I. G. Ray, R. Lu, and M. Rajarajan, "Privacy-preserving iVector based speaker verification," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 3, pp. 496–506, Mar. 2019.
[12] F. Wang, H. Zhu, X. Liu, R. Lu, F. Li, H. Li, and S. Zhang, "Efficient and privacy-preserving dynamic spatial query scheme for ride-hailing services," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11084–11097, Nov. 2018.
[13] G. Wang, R. Lu, and C. Huang, "Pguide: An efficient and privacy-preserving smartphone-based pre-clinical guidance scheme," in *Proc. IEEE Global Commun. Conf.*, 2015, pp. 1–6.
[14] Y. Wang, X. Chen, Q. Jin, and J. Ma, "LIP3: A lightweighted fine-grained privacy-preserving profile matching mechanism for mobile social networks in proximity," in *Proc. Int. Conf. Algorithms Archit. Parallel Process.*, 2015, pp. 166–176.
[15] X. Yang, R. Lu, H. Liang, and X. Tang, "SFPM: A secure and fine-grained privacy-preserving matching protocol for mobile social networking," *Big Data Res.*, vol. 3, pp. 2–9, 2016.
[16] D. Zhu, H. Zhu, X. Liu, H. Li, F. Wang, and H. Li, "Achieve efficient and privacy-preserving medical primary diagnosis based on kNN," in *Proc. Int. Conf. Comput. Commun. Netw.*, 2018, pp. 1–9.
[17] H. Zhu, X. Liu, R. Lu, and H. Li, "EPCS: An efficient and privacy-preserving classification service query framework for SVM," *Secur. Commun. Netw.*, vol. 9, no. 11, pp. 1309–1320, 2016.
[18] H. Zhu, X. Liu, R. Lu, and H. Li, "Efficient and privacy-preserving online medical prediagnosis framework using nonlinear SVM," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 3, pp. 838–850, May 2017.
[19] H. Zhu, F. Wang, R. Lu, F. Liu, G. Fu, and H. Li, "Efficient and privacy-preserving proximity detection schemes for social applications," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2947–2957, Aug. 2017.
[20] X. Zhu, J. Liu, S. Jiang, Z. Chen, and H. Li, "Efficient weight-based private matching for proximity-based mobile social networks," in *Proc. IEEE Int. Conf. Commun.*, 2014, pp. 4114–4119.
[21] O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[22] Y. Lindell, "How to simulate it–A tutorial on the simulation proof technique," in *Tutorials on the Foundations of Cryptography*. New York, NY, USA: Springer, 2017.
[23] R. Impagliazzo and S. Rudich, "Limits on the provable consequences of one-way permutations," in *Proc. ACM Symp. Theory Comput.*, 1989, pp. 44–61.
[24] J. Kilian, "Founding cryptography on oblivious transfer," in *Proc. ACM Symp. Theory Comput.*, 1988, pp. 20–31.
[25] D. Demmler, T. Schneider, and M. Zohner, "ABY-A framework for efficient mixed-protocol secure two-party computation." in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2015.
[26] M. Blum, "How to exchange (secret) keys," *ACM Trans. Comput. Syst.*, vol. 1, no. 2, pp. 175–193, 1983.
[27] M. Rabin, "How to exchange secrets with oblivious transfer," Harvard Univ., Cambridge, MA, USA, Tech. Rep. TR-81, 1981.

Thomas Schneider is full professor for Computer Science at TU Darmstadt and heads the Cryptography and Privacy Engineering Group (ENCRYPTO). In their research, he and his team demonstrate that privacy can be efficiently protected in several applications. For this, they use methods from applied cryptography and algorithm engineering for developing protocols, tools, and software prototypes to efficiently protect sensitive data. In 2019, he was awarded with an ERC Starting Grant. Before becoming a professor in March 2018, he was an independent research group leader at TU Darmstadt (2012–2018) and did his PhD in IT Security with distinction at Ruhr-University Bochum (2008–2011). In 2007, he did a research internship at Bell Labs, NJ, USA.

Amos Treiber is a PhD student at the Cryptography and Privacy Engineering Group (ENCRYPTO) at TU Darmstadt. His research focuses on cryptographic mechanisms for practically preserving privacy in various applications, especially within the domain of machine learning applications. Prior to starting his PhD in July 2018, he received his M.Sc. in IT-Security from TU Darmstadt in 2018 and his B.Sc. in Applied Computer Science from Heidelberg University in 2016.