Secure Key Exchange Protocol based on Virtual Proof of Reality

Yansong Gao

School of Electrical and Electronic Engineering,
The University of Adelaide, SA 5005, Australia,
yansong.gao@adelaide.edu.au,

Abstract. Securely sharing the same secret key among multiple parties is the main concern in symmetric cryptography that is the workhorse of modern cryptography due to its simplicity and fast speed. Typically asymmetric cryptography is used to set up a shared secret between parties, after which the switch to symmetric cryptography can be made. In this paper, we introduce a novel key exchange protocol based on physical hardware implementation to establish a shared secret between parties rather than relying on mathematical implementation of asymmetric cryptography. In particular, the key exchange is dependent on a new security concept named as *virtual proof of reality* or simply *virtual proof (VP)* that enables proof of a physical statement over untrusted digital communication channels between two parties (a "prover" and a "verifier") residing in two separate local systems. We firstly exploit the VP to secure key exchange and further prove it by using experimental data. The key transfered in this protocol is only seen by the prover and hidden from not only the adversary but also the verifier. While only the verifier can successfully discover it.

Keywords: key exchange, virtual proof, physical unclonable function (PUF), sensor PUF.

1 Introduction

Cryptographic applications require securely key exchange between parties before a secure communication channel is set up. The asymmetric crypto-system is usually used to secure a shared key (private key) exchange. However, in this case a strong random number generator is always needed, otherwise an attacker will be able to guess the private key. In general, the asymmetric cryptography is built on mathematical framework, while the strong random number generation is relying on hardware device that is capable of generating true random number rather than pseudo-random number built on mathematical means.

In this paper, we pure relying on hardware implementations to secure transfer a secret key between parties. Our work extends the work of sensor PUF [3] and VP [4]. The sensor PUF is a variant PUF that co-mingles sensing with challenge-response processing. Where a PUF [6] can be simplify treated as a black box

that its response (output) is a complex physical function of its challenge (input). The complex physical function exploits minute process variations during device manufacturing that is irreversible, uncontrollable and unclonable in physical way. The sensor PUF makes sensor and crypto module inseparable by merging sensing with cryptography, which ensures authenticity and veracity of measurements in a untrusted remote environment. Recently, Ruhrmair et. al. [4] generalize the application of sensor PUF and named it as VP enabling that the prover situated in untrusted environment proves a physical statement of a witness object (WO) to verifier through an untrusted communication channel. We extend these two works further to secure key exchange. In this work [4], the used PUF for VP proof is generalized as (WO) that is prepared by the verifier and handed over to the prover through an untrusted supply chain prior to the VP application.

2 Structure

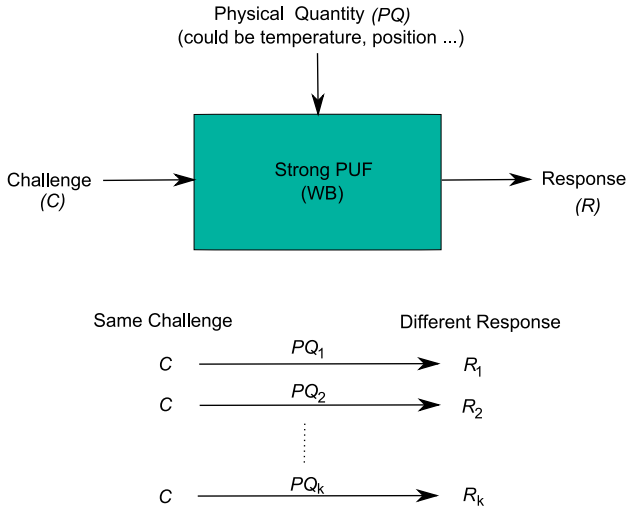

Fig. 1. Response (R) of the PUF is a function of the Challenge (C) and Physical Quantity (PQ). For the same C, different R are produced due to the difference of PQ.

To achieve securely key exchange based on VP, firstly, an WO is needed. in this paper, we deploy a strong PUF [5] as an necessary WO shown in Fig .1. This strong PUF will be transfered between the prover and verifier. Secondly, the PUF will satisfy some features:

1. The strong PUF is PQ (temperature or position) dependent in its behavior. Specifically, the response R_j^i of the strong PUF is not only a function of the challenge C_j, but also a function of its PQ_i. In other words, R_j^i =

$F_{\mathrm{PUF}}(C_j, PQ_i)$. Very different R is desirable for the same C under different discretized PQ_i.
2. The R is insensitive to other variations except the specific PQ. For example, if the specific PQ is temperature, then the strong PUF should be stable against voltage variations.
3. The PUF is resistant to model building attacks, which means that knowing many $R_j^i = F_{\mathrm{PUF}}(C_j, PQ_i)$ for various C_j and PQ_i, an adversary can not predict the unmeasured R_r^s for new $C_r \neq C_j$ or new $PQ_s \neq PQ_i$. This is actually the key feature of a strong PUF.

3 Protocol

The protocol divides into two phases: enrollment phase and key exchange phase.

3.1 Enrollment Phase

1. The verifier prepares a strong PUF_A that is depend on a specific PQ.
2. for $i = 1 : k$
   set strong PUF under PQ_i
   for $j = 1 : m$
   randomly select C_j^i, apply C_j^i to the strong PUF and measure R_j^i;
   end
   end
3. The challenge response pairs (CRPs) database is created and saved. Here $DB = C_j^i, R_j^i, PQ_i$ for $i = 1, ..., k$ and $j = 1, ..., m$.
4. The PUF_A is transfered to the prover.

3.2 Key Exchange Phase

Notably that the PQ_i can be encoded as different digital values while the encoding scheme is public. So once the verifier discover the PQ_i, he/she can figure out the corresponding key to it. For example, if PQ stands for temperature, then $T_1, T_2, ..., T_k$, where $k = 8$, can be encoded as 000, 001, 010, 011, 100, 101, 110, 111. The key exchange protocol follows three/four steps:

1. The verifier randomly picks up a C_j and sends to the prover.
2. According to the key needs to be exchanged, the prover sets PUF_A under a specific PQ_i and applies C_j to PUF_A to obtain the R_j^i. Hence, R_j^i contains the information of PQ_i that is actually the encrypted key. For example, if the key of 010 needs to be transfered, then the PUF_A is set under $PQ = T_3$. Hence, R_j^3 is acquired by the prover.
3. The verifier receives the R_j^i and compares all stored $R_j^{'i}$ with R_j^i. If one of $R_j^{'i}$ matches R_j^i, then the key encoded by PQ_i is accepted. Otherwise, if none of $R_j^{'i}$ matches R_j^i, this round key exchange is aborted. For example, the verifier receives R_j^3, then compares $R_j^{'i}$, where $i = 1, 2, ..., 8$, with R_j^3.

Only the $R_j^{`3}$ will match $R_j^3$. Then the encoded key of 010 is decrypted by the verifier. Otherwise, this round key exchange is aborted if none of $R_j^{`i}$ matches R_j^3.

4. If the transfered key is long, then the key will be divided to short length keys. Steps 1—3 will be repeated until the entire key is completely transfered.

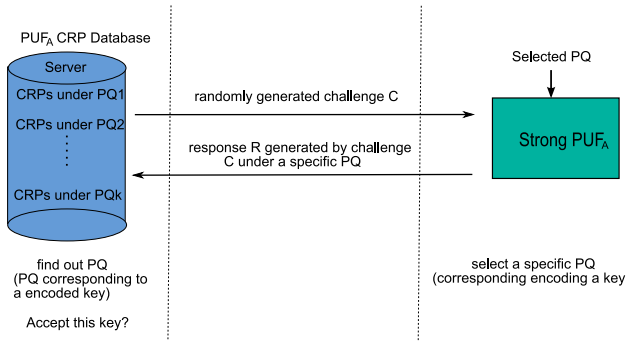

Fig. 2. Key exchange protocol.

4 Discussion and Proofs

Why this key exchange protocol is secure? Which means the adversary cannot obtain the key neither through eavesdropping nor from prediction. And the verifier can successfully discover the key encoded by the prover. The answer relies on two important pre-conditions:

1. The transfer operation of PUF$_A$ between two parties (the prover and the verifier) is secure. Even the adversary can physical access the PUF$_A$, he/she is impossible measure all of the CRPs within a period (several days, months) due to huge population of CRPs generated from the strong PUF. The adversary can not predict R for a given unused C due to its third feature in Section Structure.
2. The fractional hamming distance among all R_j^i named as *SC-DPQ-FHD* is large enough—where $i = 1, 2, ..., k$ and hence R_j^i is measured under different PQ_i to the same challenge C_j—than bit error rate (BER), shown in Fig 3 due to other variations, see the second feature in Section Structure. Where BER is FHD among R corresponding to the same C and PQ but the R evaluated multiple times. In general, BER is induced by measurement noise or other minor environment variations.

The first pre-condition ensures that the adversary can not impersonate the PUF$_A$ through a mathmatical model or create a CRP database consisting all of the

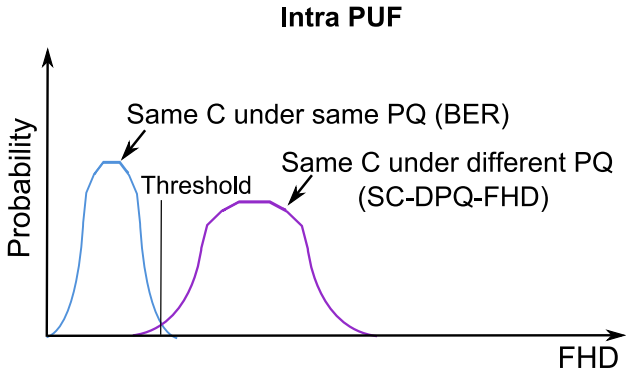

Fig. 3. Fractional Hamming Distance (FHD) Distribution. Evaluation is carried out for the same PUF.

CRPs generated from PUF_A. Hence, it ensures that the prover is the authentic part that the verifier wants to communicate with. The second pre-condition enables the verifier is capable of discovering the PQ_i on behalf of the key encoded by the prover.

To prove the pre-conditions can be met by hardware implementations and therefore show the efficiency of our proposed protocol. Experimental data in [4] is used for illustration. The PQs include temperature and position are used respectively showing different PQ of VP can be ultilised to satisfy our key exchange protocol in practice. To verify that temperature can be used as a PQ, a 4 XOR-Bistable Ring PUFs [1] (4 XOR BR-PUF) is tested. To verify that position can be treated as a PQ, the optical PUF [2] is used and tested. The performance of two key metrics that should be considered for our proposed key exchange protocol are showed in Table 1. As we can see, average SC-DPQ-FHDs are always higher than the BER for different discretized PQs. Especially when the PQ of position is used. In terms of the performance of 4 XOR BR-PUF, its performance of SC-DPQ-FHDs can be improved increase the PUF's sensitivity to temperature, which is left as our future work.

Therefore, the verifier is able to find out the encoded key (PQ_i) by the prover, even the key is hidden from both the adversary and the verifier. The adversary have no idea which key is transferred but through guessing. Simultaneously, successfully key discover by the verifier is ensured by the large FHD difference between the BER and SC-DPQ-FHD.

Table 1. average BER and SC-DPQ-FHD performance under different PQs

PQ	BER	SC-DPQ-FHD	WO
Temperature	1.0 %	6.2 %	4 XOR BR-PUF
Position	8.7 %	36.3 %	Optical PUF

5 Conclusion

In this paper, we extend the sensor PUF and VP to protect key exchange from attacking based on hardware implementation rather than number theory. The novel key exchange protocol is proposed and proved by using experimental data. This key exchange protocol can be broaden among multiple parties once more WOs are used and transfered among multiple parties, while each pair of parties is treated as the prover and verifier demonstrated in this paper.

References

1. Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, and U. Ruhrmair. The bistable ring puf: A new architecture for strong physical unclonable functions. In *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*, pages 134–141. IEEE, 2011.
2. R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.
3. K. Rosenfeld, E. Gavas, and R. Karri. Sensor physical unclonable functions. In *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*, pages 112–117. IEEE, 2010.
4. U. Rührmair, J. Martinez-Hurtado, X. Xu, C. Kraeh, C. Hilgers, D. Kononchuk, J. J. Finley, and W. P. Burleson. Virtual proofs of reality and their physical implementation. In *36th IEEE Symposium on Security and Privacy*, 2015.
5. U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber. Modeling attacks on physical unclonable functions. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 237–249. ACM, 2010.
6. G. E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th Annual Design Automation Conference*, pages 9–14. ACM, 2007.