A Chosen Random Value Attack on WPA3 SAE authentication protocol

Sheng Sun
Huawei Technologies Canada
robsun2005@gmail.com

***Abstract*— SAE (Simultaneous Authentication of Equals), is a password authenticated key exchange protocol, which is designed to replace the WPA2-PSK based authentication. The SAE Authentication Protocol supports the peer to peer (P2P) authentication, and is a major authentication mechanism of the Authentication and Key Management Suite (AKM). The SAE key exchange protocol and its variants, i.e, the Dragonfly key exchange protocol, have previously received some cryptanalysis, in which the authors pointed out Dragonfly protocol is vulnerable to the sub-group attack. This paper investigates some further vulnerabilities using impersonation attacks and suggests some protocol amendments for protection. It is recommended that SAE implementations should be upgraded to ensure protection against these attacks.**

I. INTRODUCTION

Simultaneous Authentication of Equals[1] is a password authenticated key exchange protocol specified for use within IEEE 802.11 systems. In this paper, we simply call it the SAE authentication protocol. The SAE authentication protocol is based on the Dragonfly[2] protocol and employs discrete logarithm cryptography to perform an efficient key exchange. It performs mutual authentication using a password or a pre-shared key (PSK) between two authenticating parties. The SAE protocol was originally designed to support the P2P authentication and to replace the WPA2-PSK based authentication. The SAE authentication protocol supports both the Finite Field Computation (FFC) and Elliptical Curve Computation (ECC). These give the SAE authentication protocol the benefits of reusing the safe Diffie-Hellman MODP (Modular Exponential) groups [9] specified in Internet Key Exchange protocol (IKE) [10] or the safe elliptical curves, such as NIST P-256 specified in [11]. The SAE authentication protocol also possesses the security property of Perfect Forward Secrecy (PFS) [12] useful for scenarios, such as IoT, Hotspot, etc.

The SAE protocol has a number of components. One of these is the Dragonfly protocol which is designated within the Internet Engineering Task Force (IETF) as a candidate standard for IPsec, Transport Layer Security (TLS) and Extensible Authentication Protocol (EAP) applications[6]. The SAE protocol is resistant to active attacks or online dictionary attacks, passive attacks or off-line dictionary attacks. The resistance to online dictionary attacks assures that an adversary cannot gain any advantage even by launching an active attack within key exchange procedure. Another technique with the SAE authentication protocol employs the Zero Knowledge Proof (ZKP)[8] to prevent the leak of domain parameters during the handshakes. There have been two independent cryptanalysis studies of the Dragonfly key exchange protocol [3,4]. In [3], Clarke and Hao pointed out that the Dragonfly protocol potentially is subject to sub-group attacks due to the lack of Public Key Validation. However to validate a public key would require a full exponentiation over the finite group which significantly decreases the protocol's efficiency. In [4], the authors proves that the Dragonfly protocol can achieve the Forward Secrecy in Random Oracle Model (ROM) and as secure as SPEKE protocol. ‘

In this paper it is further shown that SAE authentication protocol is also vulnerable to impersonation attacks. These could enable an adversary to bypass the authentication, by simply choosing a weaker random value during the key exchange protocol.

This paper, further reviews the security properties of the SAE authentication protocol, and outlines some additional attacks using the chosen random value impersonation attack methodology. At the end of this paper, are outlined some of the potential protection schemes for these attacks. Section II outlines the SAE authentication protocol and its known attacks. Section III illustrates further impersonation attacks on SAE based on the choice of weak keys during the protocol exchange. Section IV discusses implications of the impersonation attack. Finally, section V outlines examples of additional measures that may be employed in SAE authentication protocol to protect against these attacks. It is recommended that future implementations make provision to guard against these attacks and that installed systems be upgraded with new protection.

II. THE SAE AUTHENTICATION PROTOCOL

This section provides a brief outline of the mechanism of the basic Simultaneous Authentication of Equals protocol. SAE authentication protocol is based on discrete logarithm cryptography and the Zero Knowledge Proof (ZKP) system. An implementation of SAE authentication protocol can either use operations on a finite field or an elliptic curve group. No assumptions are made about the underlying group, other than that the computation of discrete logarithms is sufficiently computationally difficult for the level of security required. In SAE, there are two operations that can be performed: an element operation that takes an input of two elements and outputs a third element, and a scalar operation that takes an input of an element and a scalar and outputs an element. This password mapping to password element function is sometimes referred to as ”hunting and pecking”.In this process, the password is mapped into a group element. After a Password Element (PE) is generated at the ”hunting

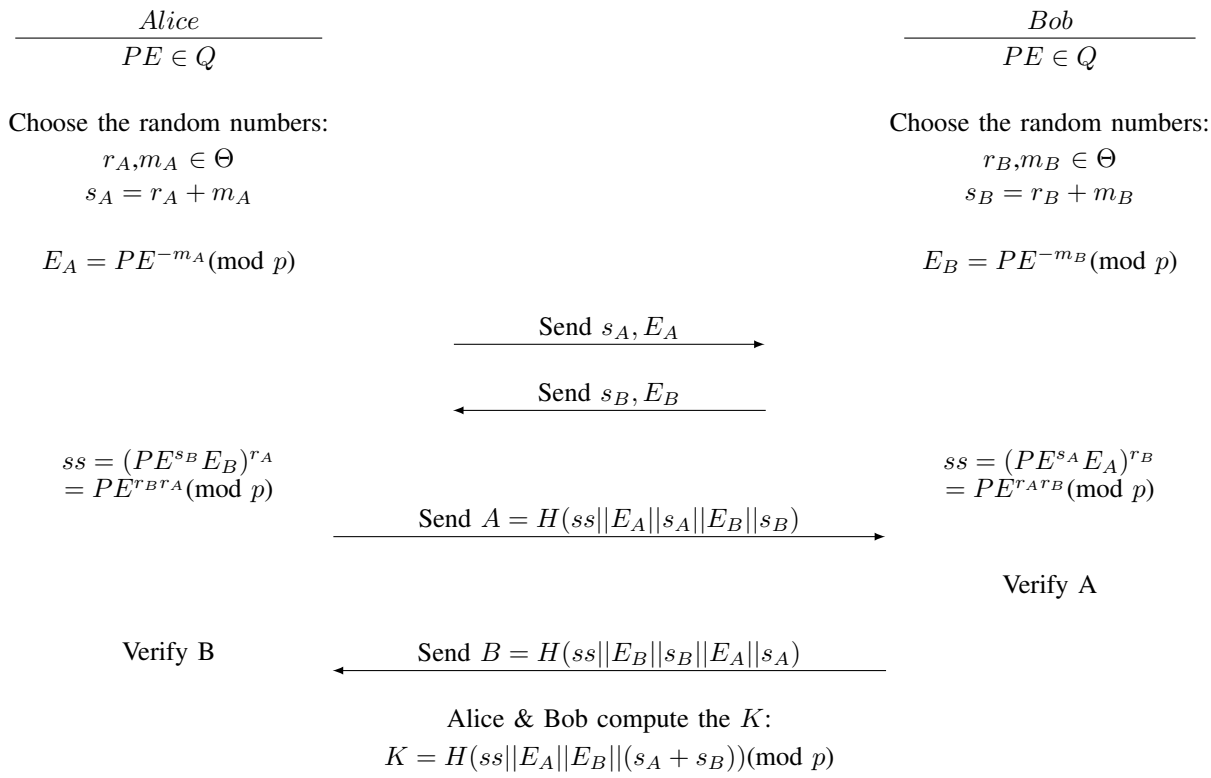

Fig. 1. SAE Authentication Protocol

and pecking" phase, both parties engage in the Dragonfly key exchange protocol to derive the session keys.

A. Outline of SAE authentication protocol

We take the finite field as an example. Let us define p as a large prime. We denote a finite cyclic group Q, which is a subgroup of Z_P^* of prime order q. Hence: $q \mid p-1$. We denote the element operation A, B for elements A and B, and the scalar operation $b \cdot A$ for element A and scalar b. These notations are in line with those commonly used when working over a finite field.

The SAE authentication protocol requires both participants to have an identical representation of the password/passphase. $Alice$, and Bob have a shared password from which each can deterministically generate a password element $PE \in Q$. The algorithms to map an arbitrary password to an element in Q are specified in [1] and [2]. The generation of the PE however is not within our attack and we will not repeat the procedure here.

A brief outline of the SAE protocol is as follows:

$Alice$ randomly chooses two variables, scalars r_A and m_A from a random space $\Theta : \{1, ., q\}$, calculates the scalar s_A = r_A + m_A (mod q) and the element E_A = P^{-m_A}(mod p) and sends s_A and E_A to Bob.

Bob randomly chooses two variables, scalars r_B, m_B from the same random space $\Theta : \{1, ., q\}$, calculates the scalar s_B = $r_B + m_B$ (mod q) and E_B = P^{-m_B}(mod p) and sends s_B and E_B to $Alice$.

$Alice$ calculates the shared secret $ss = (PE^{s_B} E_B)^{r_A}$ = $PE^{r_B r_A} (\text{mod } p)$.

Bob calculates the shared secret $ss = (PE^{s_A} E_A)^{r_B}$ = $PE^{r_A r_B} (\text{mod } p)$.

$Alice$ sends $A = H(ss||E_A||s_A||E_B||s_B)$ to Bob where H is a predefined cryptographic hash function.

Bob sends $B = H(ss||E_B||s_B||E_A||s_A)$ to $Alice$ where H is the same hashing function as Alice's.

$Alice$ and Bob check that the hashes are correct and if they are then $Alice$ and Bob can create a shared key $K = H(ss||E_A||E_B||(s_A + s_B)) (\text{mod } p)$.

This SAE authentication protocol is illustrated in Figure 1.

The SAE authentication protocol at this stage has successfully generated the shared key K which can then be fed into the Pairwise Master Key (PMK) for the subsequent IEEE 802.11 four way handshake key installation.

III. A Chosen Random Value Attack on SAE key exchange protocol

In this section,we demonstrate an attacking method which enables an adversary, identified as Eve but impersonating as Bob in the protocol, to compromise the SAE key exchange by carefully choosing the random value, r_B and m_B.

A. Attack Methodology

The new attack is a form of a chosen random value attack. The SAE protocol was carefully designed in order to thwart dictionary attacks by utilizing groups containing

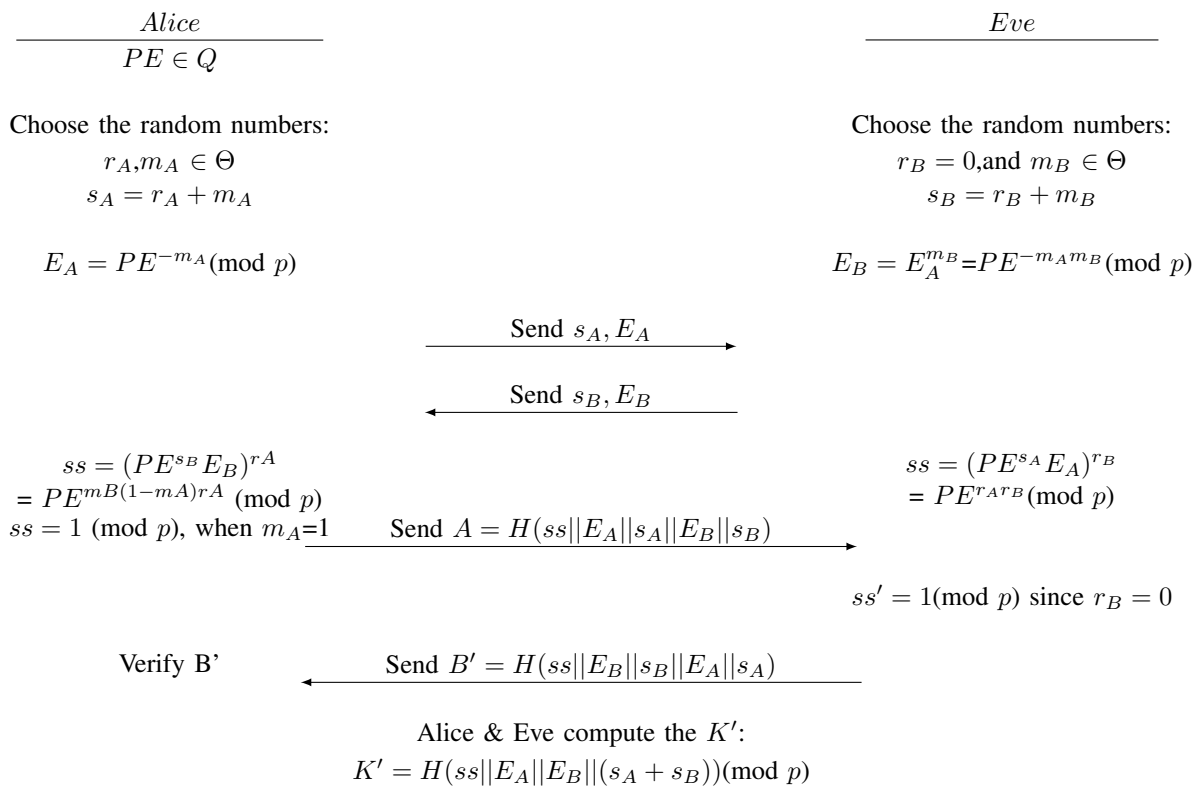

Fig. 2. The chosen random value attack on SAE protocol

small subgroups that obfuscate the group domain parameters during the handshakes [2]. The group is only identified by an the group identifier that both participants agree to use. However the SAE authentication protocol fails to fully specify how the random values, r_A, m_A, r_B, m_B must be chosen. The protocol simply specifies them as randomly generated from random space Θ. We observe that, with some probability, an adversary, Eve, is able to impersonate Bob during the SAE authentication by choosing weaker random values.

It is feasible for the adversary, Eve impersonating Bob, to carefully choose the random values r_B and m_B during the initialization of the SAE key exchange protocol. If Eve has knowledge of the proper group domain parameters to fully implement the SAE authentication protocol, Eve only lacks knowledge of the pre-shared credentials of the valid party. In 2014, Clarke and Hao [3] properly pointed out a type of impersonation attack which could take advantage of the SAE protocol's lack of public key validation during the key exchange. To address this attack, the authors in [3] recommended a remedy of performing a group element validation when receiving E_B. In our attack model we assumed the SAE authentication protocol has been extended to include the remedy in [3] even though it could be a costly solution.

In the further attack, it is assumed that $Alice$ and Bob share a common password and Eve is able to eavesdrop all the SAE protocol conversation and hijacks the session by spoofing $Bob's$ identity which is transparent during the authentication protocol, In effect, Eve may impersonate Bob becoming "Rob". Without loss of generality, we assume $Alice$ initiates an SAE session with Bob by sending S_A and E_A to Bob. When Eve intercepts the message during the transmission, it is feasible for Eve to intentionally respond with $r_B = 0$, and with m_B chosen arbitrarily from the random space Θ. The adversary's choice of $r_B = 0$ is masked by the obfuscating techniques of $s_B = r_B + m_B$ (mod q) specified in the SAE key exchange protocol. Thus the carefully chosen $r_B = 0$ won't be detectable by $Alice$ even assuming $Alice$ enforces the public key validation at the time of reception.

One challenge with Eve is how to construct the E_B given Eve has no chance to guess the pre-shared password between $Alice$ and Bob, hence no way to guess the Password Element during the "hunting and pecking" phase. However, the adversary, Eve can take advantage of the fact $Alice$ first shared the s_A and E_A with Bob, which means the Eve can obtain the value of E_A and s_A. An approach would be that Eve could reconstruct E_B by setting E_B=$E_A^{m_B}$=$PE^{-m_A m_B}$ (mod p). At the end of construction, Eve launches the attack by sending the s_B and E_B to $Alice$ (in effect, spoofing the Bob). After receiving s_B and E_B, $Alice$ then computes the ss = $(PE^{s_B} E_B)^{rA}$ = $PE^{m_B(1-m_A)r_A}$(mod p) which is then used in computation of the verification value A. At $Eves$ side, Eve just computes the ss'= $PE^{r_B r_A}$ (mod p) =1(mod p) and then constructs the verification value

$B' = H(ss||E_B||s_B||E_A||s_A)$ where the $ss = 1$ (mod p).

Without loss of generality, depending on how $Alice$ randomly chose the r_A, assuming with certain probability of p, the $r_A = 1$ was chosen or due to implementation flaw, this will cause the $Alice$ to obtain the ss = $(PE^{s_B}E_B)^{rA}$ = $PE^{mB(1-mA)rA}$ (mod p)=1 (mod p).

Thus, we can say that with probability of p, Eve will bypass the SAE authentication and successfully launch an impersonation attack. As we have pointed out earlier, the chances of bypassing the SAE authentication lies in the possibility of Alice choosing the $m_A = 1, p = 1/q$, assuming m_A is generated uniformly from a random function.

This chosen random value attack on SAE authentication protocol is illustrated on figure 2.

B. Extension of chosen random value attack with dictionary

Similarly, the chosen random value attack could be extended to a dictionary type of attack if the random space is relatively small.

If the adversary, Eve could let E_B be equal to the square root of E_A over the group Q, which calculates the E_B= $|\sqrt{E_A}|$=$|PE^{(-m_A)/2}|$ (mod p). To make sure there exist square roots of E_A over Q, it's imperative to check if the received E_A is quadratic residue over Q by checking the legendre symbol, specifically check if $(\frac{E_A}{p})$ = 1. If E_A is a quadratic residue over p, Eve will proceed with performing the off-line random value guessing of the scalar equivalence which is from the random space Θ. The well chosen random value and E_B can be used to obtain the intermediate key ss = $(PE^{s_B}E_B)^{rA}$ = $PE^{(m_B - m_A/2)r_A}$ (mod p), with $r_B = 0$. There exist many pairs of m_B and $m_A \in \Theta$ that could result in the $ss = 1$(mod p) at $Alice$. We hereby denote the pairs as scalar equivalence. Assuming the scalar equivalence exist, $Alice$ will obtain the $ss = 1$ (mod p). When $Alice$ transmits the s_A and m_A to Bob, The adversary, Eve launches the dictionary attack by finding the scalar equivalence in Θ that can obtain the intermediate key $ss = PE^{(m_B - m_A/2)r_A}$(mod p)=1 (mod p). Then Eve sends the s_B and m_B to Alice. With the expectation that $Alice$ will obtain the $ss = 1$(mod p). She computes:

$$m_B - m_A/2 = 0 \tag{1}$$

since Eve received the $s_A = r_A + m_A$, hence

$$m_B - (s_A - r_A)/2 = 0 \tag{2}$$

$$s_A = 2 \cdot m_B + r_A(modq) \tag{3}$$

Assuming the random space Θ is not sufficiently large, in order to find the scalar equivalence $\{m_B, m_A\}$, Eve could take advantage of equation (3) to build an off-line dictionary to find the proper $\{m_B, r_A\}$ pairs which are also the scalar equivalence. Then Eve sends the s_B and m_B to $Alice$. With expectation of $Alice$ to obtain the $ss = 1$ (mod p), Eve calculates the $B' = H(ss = 1||E_B||s_B||E_A||s_A)$ and transmits to $Alice$ for verification. Eve runs the dictionary attack of algorithm 1:

Algorithm 1 Extension chosen random value attack on SAE protocol

Input : s_A,m_B,r_A,m_A,$r_B = 0$
Output: B', K'

while *each* m_B *and* r_A *in* Θ **do**
  find the pairs m_B, r_A which satisfy:
  $s_A = 2 * m_B + r_A$(mod q)
  return m_B,r_A as new dictionary
end
while *each* m_B *in dictionary* **do**
  $s_B = m_B$
  E_B= $|\sqrt{E_A}|$=$|PE^{(-m_A)/2}|$(mod p)
  ss' = $(PE^{s_B}E_B)^{rA}$ = $PE^{(m_B - m_A/2)r_A}$(mod p)
  $A' = H(ss = 1||E_A||s_A||E_B||s_B)$
  if $A = A'$ **then**
    $B' = H(ss = 1||E_B||s_B||E_A||s_A)$
    $K' = H(ss = 1||E_A||E_B||(s_A + s_B))$
    return B', K'
  else
  end
end

To perform the above attack, the adversary Eve needs to find out the scalar equivalence within the random space Θ of the dictionary. As long as the size of the random space Θ is manageable, it's not difficult for an adversary to find out all the scalar equivalences in the random space. This is particularly true when the SAE authentication protocol is applied to devices using Personal Identification Numbers (PIN) or short password. These applications often chose to limit the random space in order to improve the performance.

The extension chosen random value attack above lies in computing square roots over primes, which is hard. Nonetheless, it could be argued that square roots modulo a prime p can be computed fast by Tonelli-Shanks algorithm [14].

Similar to this extension random value attack, Eve can also reuse the E_A, similar to reflection attacks, let $E_B = E_A$, then the scalar equivalence is to satisfy $s_A = m_B + r_A$(mod q). The attack algorithm is similar. This construction is much simpler to implement with less computation required.

IV. Discussion

It is observed that the SAE authentication protocol in [1] does not fully specify the random space Θ and its requirements. Unlike SPEKE protocol, SAE protocol relaxes the requirements that p must be a safe prime (i.e $p = 2 \cdot q + 1$), which is also observed in [3]. These factors enable practical attacks on SAE protocol that could result in widely deployed devices being compromised. In addition to the subgroup attack in [3], if the E_B is carefully constructed based on the received E_A then $Alice$ will not be able to detect the chosen random value attack even with the public

key validation techniques within the SAE protocol. The rationale of transmitting the scalars in the SAE authentication protocol, and the scalar construction using addition instead of other safer operations,such as multiplication is unknown. This technique may be related to obfuscating the random value r during the transmission to prevent the Man-In-The-Middle (MITM) attacks and to help the protocol efficiency. However it seems this technique actually reduces the security.

V. Preventing chosen random value attacks on the SAE Protocol

There are several ways to prevent the chosen random value attacks on SAE key exchange protocol.

1) The chosen random value attack can be prevented by implementing a validation of the intermediate key ss within the SAE authentication protocol. If $ss = 1(modp)$, then it's very likely an indication of a chosen random value attack. Such indication should cause $Alice$ to abandon the SAE authentication process.
2) There are possible other ways to construct the E_B by utilizing the intercepted s_A and E_A. It is always the best practice to implement the full public key validation as specified in NIST SP 800-56A [13], to check if the intermediate key ss belongs to the group $\in Q$. The public key validation as remedy was also recommended in [3]. Nonetheless it's well known that implementing the public key validation requires the exponentiation and would reduce the performance of the SAE authentication protocol.
3) As we have observed from the chosen value attack, it's imperative to define the random space Θ as $\{2, ., q-1\}$, instead of random space from $\{1, ., q\}$ to avoid the weak random values.
4) It is recommended to use the large random space Θ with large prime number $q \in Z_p^*$ in order to decrease the possibility of chosen random value attacks. If q is divisible by other numbers, the extension attacks could become more efficient by creating subgroups of the random space Θ.

VI. Conclusions

We demonstrated that the SAE protocol is vulnerable to a chosen random value attack and its extension attacks by computing the dictionary within the random space Θ, in which some weak random values could lead to the SAE protocol being bypassed by a third party without the knowledge of the password. These attacks are more practical to implement in scenarios where the random space Θ is relatively small, for instance, PIN or short password based environments. It might be argued that the risk of such attacks could be mitigated by anti-clogging mechanisms and by thresholding unsuccessful impersonation attempts. Nonetheless, our analysis indicates the SAE authentication protocol could be compromised by online and offline dictionary attacks. For SAE, the adversary could gain information about the random value chosen by the valid users by guessing the scalar equivalence.

Acknowledgment

The author would like to thank David Steer, Sophie Vrzic, Wen Tong, Peiying Zhu, and Rene Struik for their reviews and valuable comments.

References

[1] Simultaneous Authentication of Equals: IEEE 802.11-2012.
[2] Dan Harkins. Dragonfly key exchange - internet research task force internet draft. http://tools.ietf.org/html/draft-irtf-cfrg-dragonfly-00
[3] Clarke, D. and F. Hao, "Cryptanalysis of the Dragonfly Key Exchange Protocol", IET Information Security, Volume 8, Issue 6, DOI 10.1049/iet-ifs.2013.0081, November 2014.
[4] Lancrenon, J. and M. Skrobot, "On the Provable Security of the Dragonfly Protocol", Proceedings of 18th International Information Security Conference (ISC 2015), pp 244-261, DOI, September 2015.
[5] D. Harkins,Ed. Secure Password Ciphersuites for Transport Layer Security (TLS) draft-harkins-tls-dragonfly-02, https://wiki.tools.ietf.org/html/draft-harkins-tls-dragonfly-02.
[6] Extensible Authenticaion Protocol, IETF RFC 3748 https://tools.ietf.org/html/rfc3748
[7] David P.Jablon. Strong password-only authenticated key exchange. ACM Computer Communications Review, 26:5-26, 1996
[8] Zero Knowledge Proof, https://en.wikipedia.org/wiki/Zero-knowledge_proof
[9] T. Kivinen and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)", IETF RFC 3526, https://www.ietf.org/rfc/rfc3526.txt
[10] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)", IETF RFC 2409, https://tools.ietf.org/html/rfc2409
[11] Digital Signature Standard (DSS), NIST FIPS PUB 186-4, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
[12] Forward Secrecy, https://en.wikipedia.org/wiki/Forward_secrecy.
[13] Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-56ar2.pdf
[14] Tonelli-Shanks Algorithm, https://en.wikipedia.org/wiki/Tonelli