Addressing the Algebraic Eraser Diffie–Hellman Over-the-Air Protocol

Derek Atkins and Dorian Goldfeld

SecureRF Corporation
100 Beard Sawmill Rd #350, Shelton, CT 06484
datkins@securerf.com, dgoldfeld@securerf.com

Abstract. The *Algebraic Eraser Diffie–Hellman* (AEDH) protocol, first introduced in 2005 as a key agreement and authentication protocol, has been proposed as a standard in ISO JTC-1/SC-31 (29167-20) to protect various communication protocols like RFID, NFC, or Bluetooth for devices associated with ISO-18000 and the Internet of Things. A recent paper by M.J.B. Robshaw and Simon R Blackburn claims to recover sufficient data to impersonate a device or, with a bit more work, recover the private keys of a device if an attacker uses the draft 29167-20 protocol and gains direct access to the resulting shared secret computation. This paper shows that simply adding a Hash or a Message Authentication Code (MAC) to the proposed authentication protocol overcomes the purported attacks. These simple standard enhancements thwart all of these attacks; that is, attacks of this nature fail. As the 29167-20 draft is currently a work item under active development within the ISO process, all these attacks would normally have been addressed in the working group, and no AEDH protocol in the public domain currently transmits the computed shared secret. Therefore, contrary to the conclusion of Robshaw and Blackburn, a simple addition to the draft protocol, similar in nature to protections in other protocols like TLS, makes the AEDH protocol perfectly suitable for authentication of passive tags and other low-power, constrained devices.

Keywords: Algebraic Eraser, Group Theoretic Cryptography, E-Multiplication, Braids

1 Introduction

In 2005 I. Anshel, M. Anshel, D. Goldfeld, and S. Lemieux proposed a key agreement protocol intended for use on low-cost platforms with constrained computational resources [1]. Such platforms typically arise in passive high-frequency (HF), ultra-high frequency (UHF), active (powered) radio frequency identification (RFID)[1] tags, wireless sensors, and other resource-constrained Internet of Things devices. The protocol, called *Algebraic Eraser Diffie–Hellman* (AEDH), sometimes also called the Algebraic Eraser Key Agreement Protocol(AEKAP), has been proposed in ISO JTC-1/SC-31, project 29167-20, as an authentication scheme for various ISO 18000 protocols.

In [2] M.J.B. Robshaw and Simon R. Blackburn propose some key recovery attacks based upon a draft version of a tag authentication protocol published at [3] and still in active development within ISO. This draft protocol was developed within the confines of the ISO standardization process including several design team meetings which included M.J.B. Robshaw.

In this paper we present two straightforward changes to the tag authentication protocol that each completely defeat the key recovery attacks proposed.

[1] Robshaw and Blackburn mention RAIN RFID, which is an industry association created by one of their employers and is not part of any ISO standard or referenced by 29167-20 or any Algebraic Eraser method. Indeed, 29167-20 is suitable for devices well beyond the scope of the RAIN RFID alliance.

2 AEDH review

AEDH is a Diffie–Hellman key agreement protocol based on the Algebraic Eraser and E-Multiplication one-way function. E-Multiplication is an action that applies a generator of the Braid Group to a matrix and permutation yielding another matrix and permutation. The process of computing an AEDH shared secret involves sharing public keys comprised of a matrix and permutation and then using E-Multiplication to iterate over a private braid to compute the shared secret. A full mathematical description of AEDH can be found in [1] and is not reiterated here.

E-Multiplication is rapidly computable, which is why AEDH is so efficient and makes it a perfect candidate for low-power and constrained devices. Performance of AEDH is orders of magnitude better than ECC [4] which is yet another reason it is proposed as a security suite for the various communication protocols of ISO JTC-1/SC-31, including but not limited to RFID.

As part of the ISO standardization process an initial protocol was proposed to an SC-31 working group for the purpose of review and to validate the crypto operations. Several members of the working group were involved in reviewing and commenting on the protocol specification, which was published independently at their insistence [3]. It was this draft protocol, the output of the ISO design team, and not the AEDH method, that was attacked.

3 Key recovery attacks against the proposed OTA protocol

Out of the 15 pages of [2], the key sentence to remember is at the end of section 3.4 on page 8: "We note that all the attacks in this paper use knowledge of the shared secret key generated during the tag authentication protocol." In other words, every attack they present assumes the attacker can make several queries to the tag with invalid public keys and obtain the resulting raw computed shared secrets.

Assuming an attacker can gain access to these computed shared secrets, Blackburn and Robshaw show how the attacker can relatively quickly acquire enough information from the tag to both impersonate the tag and, with additional work, recover the private matrix. This is done by generating chosen public key data and using the tag as an oracle to produce the computation results. It should be noted that similar attacks, using invalid elliptic curves, were made against Diffie-Hellman type ECC protocols [5, 6].

These attacks work only because in the draft protocol the tag directly returns the results of the shared secret computation over the air for validation purposes.

Without direct access to the computation result the tag no longer acts as an oracle and all these attacks fail. The next section describes two practical (and straightforward) approaches that are typically implemented in the field, depending on the use case, to prevent the tag from acting as an oracle for these attacks.

4 Using a Hash or MAC to prevent Oracle attacks

There are two straightforward ways to prevent the tag from becoming an oracle. The first is to use a hash to create a validation value that does not reveal the shared secret in any way. The second is to use a nonce and Message Authentication Code (MAC) in a challenge/response protocol.

In the first approach, after computing the shared secret the tag will pass it through a hash function like SHA-1, SHA-2, or AEHash [7]. A real interrogator trying to authenticate the tag would be able to reproduce the shared secret on its own and compute the same hash. An attacker, however, would not be able to reproduce the raw shared secret without being able to reverse the hash function. This defeats all the attacks.

In the second approach, after computing the shared secret the two parties would use it to key a MAC. Either an HMAC or CMAC can be used. The interrogator sends a random nonce to the tag; the tag replies with the MAC of the nonce value. Access to the nonce and MAC value does not reveal any information about the computed shared secret. This defeats all the attacks.

5 Conclusion

Robshaw and Blackburn claim to have found invalid public key attacks which are able to recover secret data when an attacker has access to the shared secret computed using AEDH. We have shown that with a simple addition to the draft protocol we can prevent these oracle attacks. With the simple defeat of these oracle attacks the Algebraic Eraser remains be a strong candidate for authenticating low-powered and constrained devices like Bluetooth and those found in ISO 18000 including NFC, passive RFID, and active RFID.

With the simple modifications to the ISO 29167-20 draft protocols, the adding of a hash or MAC, all the attacks presented are completely defeated.

References

1. Anshel, Iris; Anshel, Michael; Goldfeld, Dorian; Lemieux, Stephane, *Key agreement, the Algebraic Eraser^{TM}, and Lightweight Cryptography,* Algebraic methods in cryptography, Contemp. Math., vol. 418, Amer. Math. Soc., Providence, RI, 2006, pp. 1–34.
2. Blackburn, Simon R.; Robshaw, M.J.B.; *On the Security of the Algebraic Eraser Tag Authentication Protocol,* http://eprint.iacr.org/2016/091 (2016).
3. SecureRF Corporation, *Algebraic Eraser OTA Authentication,* http://www.securerf.com/wp-content/uploads/2015/10/Algebraic_Eraser_Over-the-Air_Authentication.pdf.
4. Atkins, Derek *Algebraic Eraser: A lightweight, efficient asymmetric key agreement protocol for use in no-power, low-power, and IoT devices,* http://csrc.nist.gov/groups/ST/lwc-workshop2015/papers/session8-atkins-paper.pdf.
5. Antipa, Adrian; Brown, Daniel; Menezes, Alfred; Struik, René; Vanstone, Scott; *Validation of Elliptic Curve Public Keys,* Public Key Crypto, 2003, https://www.iacr.org/archive/pkc2003/25670211/25670211.pdf
6. Jager, Tibor; Schwenk, Jø"rg; Somorovsky, Juraj; *Practical Invalid Curve Attacks on TLS-ECDH,* Computer Security – ESORICS 2015, LNCS Volume 9326, pp. 407-425.
7. Anshel, Iris; Atkins, Derek; Goldfeld, Dorian; Gunnells, Paul E.; *A Class of Hash Functions Based on the Algebraic Eraser,* 2015.