International Journal of Pure and Applied Mathematics
Volume 116 No. 10 2017, 253-258
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

# MULTI-USER SCHNORR SECURITY IN NEXT GENERATION

B.Sundarraj
Assistant Professor, Department of CSE, BIST, BIHER,
BharathUniversity,Chennai.
sundarrajboobalan@gmail.com

**Abstract:** Three recent proposals for standardization of next-generation ECC signatures have included key prefixing modifications to Schnorr's signature system. Bernstein, Duif, Lange, Schwab, and Yang stated in 2011 that key prefixing is an inexpensive way to alleviate concerns that several public keys could be attacked simultaneously. Theorem by Galbraith, Malone-Lee, and Smart states that, for the classic Schnorr signature system, single-key security tightly implies multi-key security. Struik and then Hamburg, citing this theorem, argued that key prefixing was unnecessary for multi-user security and should not be standardized. This paper identifies an error in the 2002 proof, and an apparently insurmountable obstacle to the claimed theorem. The proof idea does, however, lead to a different theorem, stating that single-key security of the classic Schnorr signature system tightly implies multi-key security of the key-prefixed variant of the system. This produces exactly the opposite conclusion regarding standardization.

## 1. Introduction

For certain types of discrete logarithm based systems, which we dub Schnorr-like signature schemes, we have shown that the security does not decrease with the number of users. |Galbraith{Malone-Lee{Smart, 2002) For Schnorr and ECDSA type schemes, one does not need to include the public key in the signing process, since security in the multi-user setting is roughly the same as in the single-user setting.[1-6]

Most of the literature on signature-system security focuses on the problem of forging messages under one targeted public key. However, the real-world attacker actually sees many public keys, and presumably is happy if he solves the problem of forging messages under any one of those keys. (For example, he wants to take over somebody's account to steal money, gain computer power, relay more attacks, etc.) If the attacker has a noticeable probability of solving this problem, then the signature system is protecting most users but not all users, and it is difficult to argue that the signature system should be considered to be secure.

An attacker, who solves the first problem, namely forging messages under one targeted public key, can also solve the second problem in the same time and with the same success probability: the attacker simply targets the first key and ignores the remaining keys. The multi-key problem is therefore no harder than the single-key problem [6-10].

### *1.1 Previous work:*

This issue was directly tackled 13 years ago in a short paper by Galbraith, Malone-Lee, and Smart (henceforth \GMLS"). That paper has two main results. The first result states that, for any signature system, attacking N keys at once cannot increase the attacker's success probability by a factor above N. This has an easy proof, which works as follows.

Say we have an N-key attack that, with probability p, successfully produces a forgery. Consider the following 1-key attack:

_ Generate a list of N - 1 new key, by applying the standard key-generation procedure N - 1 times.
_ Insert the target key into the list at a random position.
_ Run the N-key attack.
_When the N-key attack asks for a signature under the target key, apply the target signing oracle.
_When the N-key attack asks for a signature under any of the N -1 new keys, apply the standard signing procedure.
_If the N-key attack successfully produces a forgery, and this forgery is under the target key, then output the forgery.

The cost of the 1-key attack is practically identical to the cost of the N-key attack. The success probability of the 1-key attack cannot be smaller4 than p=N.

In other words, the success probability of the N-key attack cannot be larger than N times the success probability of the 1-key attack [11-15].

## 2. Standardization

Internet standards, such as the TLS standard used tosecure HTTPS connections, are maintained by the Internet Engineering Task Force (IETF). IETF delegates research questions to its research arm, the Internet Research Task Force (IRTF), and in particular delegates cryptographic questions to part of IRTF, the Crypto Forum Research Group (CFRG). For the last 18 months, about 3000 messages, the primary topic on the CFRG mailing list has been elliptic-curve cryptography. For the last 6 months, about 700 messages, the primary topic has been elliptic-curve signatures. There have been five specific proposals of signature systems and several \tweaks" of those proposals.

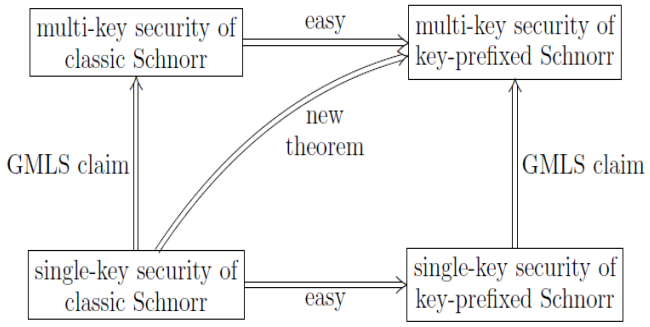
In a CFRG message dated 4 September 2015, Struik recommended against key prefixing. Ideally, signing should be possible without requiring the signer to access its public key", he wrote; and, finally bringing security proofs into the discussion, he cited the GMLS paper to conclude that one does not need to include the public key in the signing process". Hamburg echoed Struik'srecommendation, saying that \there is little gain" to key prefixing in light of the papers he cited".

This story is obviously a triumph for provable security: a tight standard-model security theorem is used to simplify and streamline cryptography that is being standardized for real-world applications. Some designers guessed that key prefixing would be helpful for security; but the choice of a standard is guided by theorems rather than by guesses. The GMLS theorem shows that the intended security goal is achieved without this unnecessary design element, so the design element is eliminated, the same way that HMQV provably dispenses" with some elements of MQV [16-18].

**Contributions of this paper:**

This paper identifies an error in the GMLS proof, and an apparently insurmountable obstacle to the claimed theorem. On the positive side, this paper shows that the attacker's probability of breaking any 1 of N signature keys in the key-prefixed variant of the Schnorr system is at most the attacker's probability of breaking a targeted signature key in the classic Schnorr system. In other words, anyone who believes that the classic Schnorr system is secure in the single-key case also has to believe that the key-prefixed variant of the system is secure in the multi-key case [19].

If the theorem claimed by GMLS were proven then it would trivially imply this new theorem, by two different proofs, where one proof moves vertically and then horizontally in the following diagram, and the other proof moves horizontally and then vertically:




However, the vertical implications in this diagram are actually unproven and presumably unprovable, leaving the new diagonal theorem as the only known way to move from single-key security to multi-key security.

**Standardization, revisited:**

After the announcement of this new theorem(and of the obstacle to the originally claimed theorem), CFRG settled on a signature system that includes key prefixing. One can still characterize this story as a triumph for provable security: cryptography equipped with a tight standard-model security theorem is standardized for real-world applications, in preference to slightly more streamlined cryptography that is not equipped with such a theorem. But what is particularly interesting about the complete story is the reversal of recommendations: one moment provable security is being used to justify a recommendation against key prefixing, and the next moment provable security is being used to justify a recommendation for key prefixing.

In retrospect the first recommendation can and should be dismissed as not being a valid example of provable security. There was a small error in the proof; this error led to an erroneous claim of a theorem, which in turn led to an unjustified recommendation regarding standardization. But this avalanche of errors raises a much larger question: what protection does the cryptographic standardization process have against errors in provable-security claims?

For comparison, the security impact of errors in programs is one of the central topics in the security literature. The unfortunate reality is that user security can be, and frequently is, compromised by accidental (or malicious) programming errors anywhere in a very large trusted code base. Simply fixing each bug as it isdiscovered obviously does not produce a secure system, whereas there is at least some hope of obtaining a secure system through systematically (1) reducing the amount of trusted code and (2) eliminating errors in the code that remains.

There is similarly widespread awareness that some provable-security claims have errors, but the primary response consists of fixing each error as it is discovered. There has been relatively little effort to understand and systematically mitigate the security impact of a continuing series of errors.

**A generalization of the Schnorr signature system:**
This section presents a parameterized family of signature systems that includes (1) the classic Schnorr signature system and (2) the key-prefixed variant of the system.

**Signature systems**:
A signature system, by definition, consists of four sets and three algorithms satisfying one condition. The four sets are a set k of secret keys", a set K of public keys", a set M of messages", and a setof signatures". The three algorithms are as follows:
_Gen (“key generation"). Inputs: none. Outputs: a public key and a secret key.
_ Sign (“signing"). Inputs: a message and a secret key. Output: a signature.
_ Verify (“verification"), required to be deterministic. Inputs: a message, a signature, and a public key. Output: True or False.
The condition is that Verify(M;_;K) = True whenever Gen() = (K; k) and Sign(M; k) = _; i.e., verification accepts any signature produced by signing, under any keys produced by key generation. Beware that several variations of this definition appear in the literature. Sometimes Sign is allowed to maintain state, i.e., to output a new secret key that is then used as a replacement for the original secret key; see, e.g., the original Goldwasser {Micali{Rivest definition of signature systems, or more recent papers on hash-based signatures. Sometimes Sign is allowed to fail after a specified number of signatures. Sometimes Sign and Verify are allowed to abort, begging the question of what a user is supposed to do for messages that trigger the abort. Sometimes Gen takes a “security parameter" as input. Sometimes cost limits are placed on Gen, Sign, and Verify as part of the definition, rather than as part of subsequent cost analyses.

## 3. Invalid inputs

In typical formalizations of algorithms, inputs and outputs are required to be strings, so the sets above are also required to be sets of strings. People defining signature systems conventionally assume that the inputs are in the specified sets. This leaves unspecified what the algorithms do if inputs are “invalid", i.e., not in the specified sets.

**Key prefixing:**
Key prefixing is a generic transformation that converts any signature scheme into what I call a “key-prefixed" signature scheme. In short, a key-prefixed signature scheme inserts the public key in front of the message before signing or verifying it.

**Security:**
This section builds a 1-key attack ReRandomizeN;Tag(A) against the classic Schnorr signature system SchnorrG;B;H, using an N-key attack A against the key-tag-prefixed system Tag+SchnorrG;B;H. The cost of ReRandomizeN;Tag(A)is practically identical to the cost of A, and Theorem 3.6 states that if the Tag function is injective on G then the success probability of ReRandomizeN;Tag(A) equals the success probability of A. In other words, key-prefixed Schnorr is at least as secure against multi-key attacks as the classic Schnorr system is against single-key attacks.
The special case ReRandomizeN;Empty(A) is, aside from irrelevant details, exactly the construction given by GMLS. However, in the same special case, the probability statement is wrong. Section 4 gives an example of an attack A for which ReRandomizeN;Empty(A) has success probability 0 while A has much larger success probability.

**Single key security of signature systems:**
A 1-key attack A against a signature system X receives two inputs: first, a public key K; second, an oracle for M 7! Sign(M; k). Here (K; k) is output by Gen. The attack is successful if its output is (M;_;K) where
_ M is a message,
_ is a signature,
_Verify(M;_;K) = True, and
_ M was not a query to the oracle.
Define PrForge1;X(A) as the probability that A is successful over all coin ipsin Gen, in each call to Sign, and in A itself.
I have deviated slightly from the traditional syntax for attacks: specifically, the attack is required to repeat the same public key K as part of its output. This has the notational advantage that a 1-key attack is, as one would expect, the special case N = 1 of an N-key attack, defined below.

**Multi-key security of signature systems:**
An N-key attack against a signature system X receives 2N inputs: public keys K1;K2; : : : ;KN and oracles O1;O2; : : : ;ON, where Oi is an oracle for M 7! Sign(M; ki). Here each (Ki; ki) is an independent uniform random output from Gen. The attack is successful if its output is (M;_;K0) where
_ M is a message,
_ is a signature,

_ K0 2 fK1;K2; : : : ;KNg,
_Verify(M;_;K0) = True, and
_ M was not a query to Oi for any i with Ki = K0.
Define PrForgeN;X(A) as the probability that A is successful over all coin ipsin the N calls to Gen, in each call to Sign, and in A itself.

**Time:**
There is a standard convention of counting the time for the legitimate user algorithms as part of the time of the attack (with the caveat that this disregards, e.g., the real time saved from having many users generate keys in parallel). With this convention, the time for ReRandomizeN;Tag(A) is practically identical to the time for A, under minor hypotheses regarding the costs of group operations etc.

**The importance of injectivity:**
This section analyzes the consequences for Theorem 3.6 if Tag is allowed to be non-injective on G.

**Injectivity on the set of generated keys:**
The critical step in the logic is from Tag(Kj) = Tag(Ki) to Kj = Ki. This step does not need Tag to be injective on all of G; it still works if Tag is injective on fK1;K2; : : : ;KNg.
For example, if N = 260 and t = 160, then a uniform random function Tag collides on fK1;K2; : : : ;KNg with probability at most 260(260□1)=2161 < 2□41, since there are at most 260 distinct elements of fK1;K2; : : : ;KNg. One can reasonably argue that 2□41 is acceptable as an attack probability.

**Empty tags:**
The approach above obviously breaks down for short tag lengths, and in particular for empty tags. The problem goes far beyond one step in the logic: if tags collide then the important conclusionl namely, that ReRandomizeN;Tag(A) succeeds if A succeeds lis simply wrong. As an extreme example, take empty tags, and consider the following very slow 2-key attack A:
_ Ask the _rst key for a signature of the message “yes".
_ Use the baby-step-giant-step algorithm to compute the discrete logarithm of the second key.
_ Apply the normal signing procedure to forge a signature of the message “yes" under the second key.

## 4. Generalizations
In previous sections I focused on the classic Schnorr signature system and its key- tag-prefixed variant. However, there are several reasons that modern signature systems have deviated in further ways from Schnorr's system; see generally for background. This section briefly analyzes the multi-key security of a broader class of signature systems.

## 5. Conclusion
Instead of generating r as a uniform random element of Z=`, a derandomized signer generates r as F(z;M), where z is another secret key. If the random function M 7! F(z;M) is indistinguishable from a uniform random function f0; 1g_ ! Z=`, a standard “PRF" design goal, then the derandomized signer is indistinguishable from a signer that generates a uniform random r for each M, while remembering which r was used for each M. Carrying out an attack against a derandomized signer is thus indistinguishable from carrying out a restricted attack against a classic signer, where the restriction is that the attack does not ask for multiple signatures on the same message.

In the multi-key setting, consider N signers with secrets z1; z2; : : : ; zN, where the ith signer generates r as F(zi;M). Indistinguishability of these signers from classic signers, with the same restriction of asking each signer for at most one signature on each M, follows from indistinguishability of the function i;M7! F(zi;M) from a uniform random function. The latter indistinguishability is an N-key version of the standard PRF goal; it cannot be broken with probability more than N times larger than the probability of breaking the standard PRF goal. There are many standard ways to design a PRF for a very high securitylevel using a large key z, absorbing the impact of this factor of N (for any plausible size of N) and preserving the overall security of the signature system.

**Key derivation:**
Ed25519 and the more general Ed DSA actually generate (k; z) by hashing a single master secret. The indistinguishability of these outputs from independent uniform random secrets k and z is another standard pseudo randomness assumption on the hash function. As above, there is at most a loss factor of N in moving to N keys.

## References

[1] B.Sundarraj Key dispensation scheme for Wireless Sensor Network, Middle-East Journal of Scientific Research 19 (6): 780-783, 2014.
[2] NeerajMittal and Ramon Novales, 2010. Clustering Based Key Pre distribution Using Deployment Knowledge, IEEE Transaction on Dependable and Secure Computing, 7(3).
[3] HuyenThi, Mohsen Guizani and Minho Jo, 2009. An Efficient Signal Range Based Probabilistic Key Predistribution Scheme in a Wireless Sensor Network, IEEE Transactions on Vehicular Networks, 58: 5.

[4] Udayakumar R., Kaliyamurthie K.P., Khanaa, Thooyamani K.P., Data mining a boon: Predictive system for university topper women in academia, World Applied Sciences Journal, v-29, i-14, pp-86-90, 2014.
[5] Kaliyamurthie K.P., Parameswari D., Udayakumar R., QOS aware privacy preserving location monitoring in wireless sensor network, Indian Journal of Science and Technology, v-6, i-SUPPL5, pp-4648-4652, 2013.
[6] BrinthaRajakumari S., Nalini C., An efficient cost model for data storage with horizontal layout in the cloud, Indian Journal of Science and Technology, v-7, i-, pp-45-46, 2014.
[7] BrinthaRajakumari S., Nalini C., An efficient data mining dataset preparation using aggregation in relational database, Indian Journal of Science and Technology, v-7, i-, pp-44-46, 2014.
[8] Khanna V., Mohanta K., Saravanan T., Recovery of link quality degradation in wireless mesh networks, Indian Journal of Science and Technology, v-6, i-SUPPL.6, pp-4837-4843, 2013.
[9] Khanaa V., Thooyamani K.P., Udayakumar R., A secure and efficient authentication system for distributed wireless sensor network, World Applied Sciences Journal, v-29, i-14, pp-304-308, 2014.
[10] Udayakumar R., Khanaa V., Saravanan T., Saritha G., Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction, Middle - East Journal of Scientific Research, v-16, i-12, pp-1781-1785, 2013.
[11] Khanaa V., Mohanta K., Saravanan. T., Performance analysis of FTTH using GEPON in direct and external modulation, Indian Journal of Science and Technology, v-6, i-SUPPL.6, pp-4848-4852, 2013.
[12] Kaliyamurthie K.P., Udayakumar R., Parameswari D., Mugunthan S.N., Highly secured online voting system over network, Indian Journal of Science and Technology, v-6, i-SUPPL.6, pp-4831-4836, 2013.
[13] Thooyamani K.P., Khanaa V., Udayakumar R., Efficiently measuring denial of service attacks using appropriate metrics, Middle - East Journal of Scientific Research, v-20, i-12, pp-2464-2470, 2014.
[14] R.Kalaiprasath, R.Elankavi, Dr.R.Udayakumar, Cloud Information Accountability (Cia) Framework Ensuring Accountability Of Data In Cloud And Security In End To End Process In Cloud Terminology, International Journal Of Civil Engineering And Technology (Ijciet) Volume 8, Issue 4, Pp. 376–385, April 2017.
[15] R.Elankavi, R.Kalaiprasath, Dr.R.Udayakumar, A fast clustering algorithm for high-dimensional data, International Journal Of Civil Engineering And Technology (Ijciet), Volume 8, Issue 5, Pp. 1220–1227, May 2017.
[16] R. Kalaiprasath, R. Elankavi and Dr. R. Udayakumar. Cloud. Security and Compliance - A Semantic Approach in End to End Security, International Journal Of Mechanical Engineering And Technology (Ijmet), Volume 8, Issue 5, pp-987-994, May 2017.
[17] Thooyamani K.P., Khanaa V., Udayakumar R., Virtual instrumentation based process of agriculture by automation, Middle - East Journal of Scientific Research, v-20, i-12, pp-2604-2612, 2014.
[18] Udayakumar R., Thooyamani K.P., Khanaa, Random projection based data perturbation using geometric transformation, World Applied Sciences Journal, v-29, i-14, pp-19-24, 2014.
[19] Udayakumar R., Thooyamani K.P., Khanaa, Deploying site-to-site VPN connectivity: MPLS VsIPSec, World Applied Sciences Journal, v-29, i-14, pp-6-10, 2014.
[20] T. Padmapriya and V. Saminadan, “Improving Throughput for Downlink Multi user MIMO-LTE Advanced Networks using SINR approximation and Hierarchical CSI feedback”, International Journal of Mobile Design Network and Innovation- Inderscience Publisher, ISSN : 1744-2850 vol. 6, no.1, pp. 14-23, May 2015.
[21] S.V.Manikanthan and K.srividhya "An Android based secure access control using ARM and cloud computing", Published in: Electronics and Communication Systems (ICECS), 2015 2nd International Conference on 26-27 Feb. 2015,Publisher: IEEE,DOI: 10.1109/ECS.2015.7124833.
[22] Rajesh, M., and J. M. Gnanasekar. "Path observation-based physical routing protocol for wireless ad hoc networks." International Journal of Wireless and Mobile Computing 11.3 (2016): 244-257.