Efficient threshold signature, multisignature and blind signature schemes based on the Gap-Diffie-Hellman-group signature scheme

Alexandra Boldyreva*

August 12, 2002

Abstract

We propose a robust proactive threshold signature scheme, a multisignature scheme and a blind signature scheme which work in any Gap Diffie-Hellman (GDH) group (where the Computational Diffie-Hellman problem is hard but the Decisional Diffie-Hellman problem is easy). Our constructions are based on the recently proposed GDH signature scheme of Boneh et al. [BLS]. Due to the instrumental structure of GDH groups and of the base scheme, it turns out that most of our constructions are simpler, more efficient and have more useful properties than similar existing constructions. We support all the proposed schemes with proofs under the appropriate computational assumptions, using the corresponding notions of security.

Keywords: Signature schemes, threshold signatures, multisignatures, blind signatures, GDH groups.

*Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. E-mail: aboldyre@cs.ucsd.edu. URL: http://www-cse.ucsd.edu/users/aboldyre.

Contents

1 Introduction **3**
- 1.1 The new GDH threshold signature scheme. 3
- 1.2 The new GDH multisignature scheme. 4
- 1.3 The new GDH blind signature scheme. 6

2 Signature schemes and their security **6**

3 The Gap-Diffie-Hellman (GDH) groups and the GDH signature scheme **7**
- 3.1 GDH groups 7
- 3.2 The GDH signature scheme 7

4 Robust proactive threshold GDH signature scheme **8**
- 4.1 Definitions 8
- 4.2 TGS, the threshold GDH signature scheme 9
- 4.3 Adding proactive security 9

5 The GDH multisignature scheme **10**
- 5.1 Multisignature schemes 10
- 5.2 MGS, the GDH multisignature scheme 10
- 5.3 Security of multisignatures 11
- 5.4 Security of the MGS multisignature scheme 12

6 The blind GDH signature scheme **12**
- 6.1 BGS, the blind GDH signature 12
- 6.2 Security of blind signatures 12
- 6.3 Chosen target CDH assumption 13
- 6.4 Security of the BGS blind signature scheme 14

7 Acknowledgements **14**

References **14**

A Proof of Theorem 4.2 **16**

B Proof of Theorem 5.2 **17**

C Proof of Theorem 6.3 **17**

1 Introduction

Recently Boneh, Lynn and Shacham [BLS] proposed a new signature scheme that uses groups where the Computational Diffie-Hellman (CDH) problem is hard but the Decisional Diffie-Hellman (DDH) problem is easy. (Recall that the CDH problem asks to compute $h = g^{\log_g u \cdot \log_g v}$ given the three random group elements (g, u, v) and the DDH problem asks to decide whether the four group elements (g, u, v, h) are all random or they are a valid Diffie-Hellman tuple, namely, they have the property that $\log_g u = \log_v h$.) Following [BLS] we will refer to such groups as Gap Diffie-Hellman (GDH) groups. The first example a GDH group is given in [J] and more details on the existence and composition of GDH groups can be found in [JN, BF, BLS]. Another signature scheme that works in GDH groups has been proposed by Lysyanskaya in [L]. Unlike the scheme of [BLS], it does not use random oracles but is less efficient.

Let G be a GDH group of prime order p and let g be a generator of G. Similarly to most discrete-log-based schemes, the secret key of the signature scheme GS of [BLS] is a random element $x \in Z_p^*$ and the public key is $y = g^x$. To sign a message $M \in \{0,1\}^*$ a signer who holds x computes the signature $\sigma = H(M)^x$, where H is a hash function mapping arbitrary strings to the elements of $G \setminus \{1\}$, where 1 denotes the identity element of G. Following [BLS] let us denote $G^* = G \setminus \{1\}$. In order to verify the validity of a candidate signature σ of a message M, a verifier simply checks whether $(g, y, H(M), \sigma)$ is a valid Diffie-Hellman tuple.

Boneh et al. [BLS] prove that signature scheme GS is secure against existential forgery under chosen message attack in the random oracle model assuming that the underlying group is GDH. They also show that using this signature scheme in some GDH groups leads to very short signatures of length approximately 160 bits. In this paper we show that besides this attractive property, GS gives rise to various efficient extensions. More precisely, we propose a robust threshold proactive signature scheme, a multisignature scheme and a blind signature scheme which are all based on the GS signature scheme. Thanks to the elegant structure of GDH groups and of the base scheme it turns out that most of our constructions are simpler, more efficient and have more useful properties than similar existing constructions. We support all the proposed schemes with proofs under the appropriate computational assumptions using the corresponding notions of security.

1.1 The new GDH threshold signature scheme.

The idea behind the (t, n)-*threshold cryptography* approach [B, D88, DF89, S] is to distribute secret information (i.e. a secret key) and computation (i.e. signature generation or decryption) between n parties in order to remove single point of failure. The goal is to allow any subset of more than t parties to jointly reconstruct a secret and perform the computation while preserving security even in the presence of an active adversary which can corrupt up to t (a threshold) parties. A review of research on threshold cryptography is presented in [D94].

In threshold signature schemes the secret key is distributed among n parties with the help of a trusted dealer or without it by running an interactive protocol among all parties. To sign a message M any subset of more than t parties can use their shares of the secret and execute an interactive signature generation protocol, which outputs a signature of M that can be verified by anybody using the unique fixed public key. The security notion for threshold signature schemes requires that no polynomial-time adversary that corrupts any t parties can learn any information about the secret key or can forge a valid signature on a new message of its choice. An important property of threshold signature schemes is *robustness*, which requires that even t malicious parties that deviate from the protocol cannot prevent it from generating a valid signature. Another useful property of a threshold signature scheme is *proactivness* [OY, CH] (or periodic refreshment of shares of a secret) whose goal is to protect a system from an adversary that builds-up knowledge of a secret by several attempted break-ins to several locations. In general, the main goals of threshold signature constructions are to *provably* achieve the following properties: to support as high a threshold t as possible, to avoid use of a trusted dealer, to be robust, proactive and as efficient as possible in terms of computation, interaction and length of the shares.

THE NEW GDH THRESHOLD SIGNATURE SCHEME. In Section 4.2 we propose the threshold signature scheme TGS that works in any GDH group. It is based on the GDH signature scheme of [BLS], whose structure turned out to be very instrumental in constructing an efficient threshold signature scheme. Our threshold GDH group signature scheme can tolerate any $t < n/2$ malicious parties, which is an optimal result. Its key generation protocol does not require a trusted dealer. The signature generation protocol does not require interaction or any zero-knowledge proofs, and avoids other difficulties pertaining to various threshold schemes. The signature generation protocol has a minimal overhead compared to that of the base scheme. The shares are short and their length does not depend on the number of parties. Its signature share generation protocol is basically the signing algorithm of the base scheme and the signature reconstruction requires only multiplication of shares. We state the security result in Theorem 4.2. The proof is in the random oracle model only because the latter is used in the proof of security of the base signature scheme. We also show how proactive security can be added to our scheme using general methods of [HJKY, HJJKY].

RELATED WORK. There exist many threshold signature scheme constructions, i.e. [DF89, H, DF91, FD, FGMY, R, GJKR96, Sh]. The proposals of [DF89, H] lack security proofs, the schemes of [DF89, DF91, FD] are non-robust while those of [FGMY, R] are robust and proactive but require a lot of interaction. We compare our scheme with the threshold DSS signature scheme of Gennaro et al. [GJKR96] and with the threshold RSA scheme of Shoup [Sh].

The threshold DSS signature proposed in [GJKR96] is robust, does not require a trusted dealer and has a proof of security without random oracle assumption. It deals with technical difficulties such as combining shares of two secrets into shares of the product of these secrets and producing shares of a reciprocal of a secret given shares of this secret. To achieve robustness, the authors use error-correction techniques of Berlekamp and Welch [BW]. As a result, the threshold DSS can tolerate only $t < n/4$ malicious parties, the threshold signature-generation protocol requires a lot of interaction and the complexity of a threshold scheme increases considerably related to the base signature scheme. The scheme can be made proactive following the methods of [HJKY, HJJKY].

The robust threshold RSA signature scheme of [Sh] is proven secure in the random oracle model. It can tolerate $t < n/2$ malicious parties and its signature generation algorithm is non-interactive. It, however, requires a trusted dealer to run the key generation protocol. The public key uses an RSA modulus that is a product of two safe primes. The protocol utilizes zero-knowledge proofs in the random oracle model in order to achieve robustness. Proactivization is not considered in [Sh].

1.2 The new GDH multisignature scheme.

In order to gain intuition about what multisignature schemes are we first discuss this notion informally and compare it to other notions.

A multisignature scheme allows any subgroup of a group of players to jointly sign a document such that a verifier is convinced that each member of the subgroup participated in signing. The trivial solution which satisfies the above informal definition is as follows. The resulting multisignature is simply a concatenation of a description of the subgroup and of regular signatures computed by each member of the subgroup using its own secret key. In fact this simple scheme will meet the security requirements we formalize in Section 5.3. Its main drawback, however, is that the signature length and verification time grows linearly with the number of users in the subgroup.

Multisignature schemes are different from threshold signatures for several reasons. The goal of a multisignature is to prove that each member of the stated subgroup signed the message and the size of this subgroup can be arbitrary, whereas in the latter setting the goal is to prove that *some* subgroup of *sufficient* size signed the message, and the minimal size is a parameter of the scheme and should be known in advance. As opposed to multisignatures, a threshold signature does not reveal identities of individual signers. Another difference is that the verification protocol of a threshold signature scheme does not depend on the current subgroup of signers. Multisignatures are also different from group signatures [CH, CS] and ring signatures [RST], where every *individual* member of the group can produce a valid signature on behalf of the whole group. In the latter two settings a signer remains anonymous with respect to a verifier. In the group signature setting there is also a third party called a group manager which can identify the identity of the signer.

Related work. Multisignatures have been introduced in [IN] and have been the topic of many other works such as [H, LHL, HMP, OO91, OO99, MOR]. The schemes of [OO91, OO99] do not support subgroups of signers, they allow only the case where each player of the group signs the document. The solutions of [IN, O] are not very efficient: multisignature generation and verification time grows linearly with the number of players. But most importantly, until recent works of Ohta and Okamoto and of Micali et al. [OO99, MOR] there were no formal notions of security for multisignatures and therefore there were no provably secure multisignature schemes. As a result, the proposals of [LHL, H] have been successfully attacked. The notion of security of [OO99] is not strong enough since it does not consider the possibility of adversarial attacks during key generation.

Micali et al. [MOR] first formalize the strong notion of security for multisignatures (they call them "accountable-subgroup multisignatures"). They modify the Schnorr-signature-based multisignature scheme originally proposed by Ohta and Okamoto in [OO99] and prove its security. The model of security and the multisignature scheme of [MOR] assume that the subset of signers L is known a priori. Each signer has to know all participants of the current subgroup of signers L, a description of which is hashed and signed along with a message. [MOR] state it as an interesting open problem to find a provably secure multisignature scheme where the composition of the subgroup can be decided after the signature shares are computed.

The new GDH multisignature scheme. In Section 5.1 we give precise definitions of multisignature schemes and their security. Our model of security is very similar to the simplified model of security of [MOR], but it is more general, it does not have the restriction that the subset of signers should be known in advance. In Section 5.2 we propose the new GDH multisignature scheme MGS. Our MGS scheme solves the open problem stated in [MOR]: it does not require a priori knowledge of a subgroup of signers and is provably secure. We state the security result in Section 5.4 and provide a proof in Section B. Moreover, it is more efficient than the one of [MOR] which requires three rounds of communication for the multisignature generation protocol, where MGS requires only one, it is basically non-interactive. Similarly to their scheme, the signature length and verification time for MGS is independent of the size of the subgroup and is almost the same as for the base signature scheme. In fact each signature share of our multisignature scheme is the standard GDH signature. In the scheme of [MOR] a signer is not allowed to begin a new signing protocol until the previous one has completed. This is because their proof of security uses rewinding which is incompatible with concurrency. Our scheme does not have such restriction not only because our proof does not use rewinding but mostly because the signing protocol is non-interactive.

We note that the approach underlying the construction of the multisignature scheme MGS can be used to achieve efficient batch verification of GDH signatures of the same message under different public keys.

1.3 The new GDH blind signature scheme.

Blind signatures are the basic tool of digital cash schemes. Using a blind signature protocol a user can obtain from a bank a digital coin, that is a token properly signed by the bank. The goal of blind signature protocols is to enable a user to obtain a signature from a signer so that the signer does not learn information about the message it signed and so that the user cannot obtain more than one valid signature after one interaction with the signer. Chaum [C] first proposed the RSA-based blind signature scheme. However, it has been proved secure only recently by Bellare et al. [BNPS]. The reason for this time gap is that it appears impossible to prove security of Chaum's scheme based on standard RSA assumptions. The approach taken by [BNPS] is to introduce the new plausible computational assumption, namely, "chosen-target-one-more-RSA-inversion" and to prove security of Chaum's RSA blind signature based on this assumption. In [BNPS] the authors suggest that an analogue of this assumption can be formulated for any family of one-way functions.

In Section 6.1 we define the new blind signature scheme BGS that works in GDH groups. The protocol is very similar to the RSA blind signature protocol. Namely, a user multiplies hash of the message with a random group element, submits it to the bank and later "derandomizes" the signature obtained from the bank using knowledge of the public key and of the random factor. In order to prove the security of BGS we follow the approach of [BNPS] and in Section 6.3 define a new computational problem, the Chosen-target Computational-Diffie-Hellman problem: given a set $Z = \{z_1, \ldots, z_n\}$ of random members of a cyclic group $G =< g >$ of prime order p, a random public key $y = g^x$ and the "helper" oracle $(\cdot)^x$ the goal is to output a set of distinct G elements $\{v_1, \ldots, v_l\}$ such that for all $i \in \{1, \ldots, l\}$ there exists $z_j \in Z$ with $z_j = v_i^x$ while the number of queries to the helper oracle is strictly less than l. Note that without the helper oracle the Chosen-target CDH assumption is equivalent to the standard CDH assumption. We conjecture that the chosen-target CDH problem is hard for all groups where the CDH problem is hard. This includes GDH groups. In Section C we prove the security of the blind signature BGS scheme under the Chosen-target CDH assumption.

2 Signature schemes and their security

SIGNATURE SCHEMES. A signature scheme S consists of three algorithms. The randomized *key generation* algorithm $\mathcal{K}$ takes a global information I and outputs a pair (sk, pk) of a secret and a public keys. The global information can contain, for example, a security parameter, a description of the group and its generator, and the description of the hash function. We do not focus on who generates these parameters and assume that they are publicly available. A (possibly) randomized *signature generation* algorithm $\mathcal{S}$ takes a message M to sign and global info I and a secret key sk and outputs M along with a signature σ. A deterministic verification algorithm $\mathcal{V}$ takes a public key pk, a message M and a signature σ and outputs 1 (accepts) if the signature is valid and 0 (rejects) otherwise. In the random oracle model [BR] both signing and verification algorithms have access to the random hash oracle. Usually $M \in \{0,1\}^*$. The common requirement is that $\mathcal{V}(pk, \mathcal{S}(I, sk, M)) = 1$ for all $M \in \{0,1\}^*$.

SECURITY. The widely-accepted notion of security for signature schemes is unforgeability under chosen-message attacks [GMR]. We recall this notion adjusted to the random oracle model.

Definition 2.1 Let $S = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ be a signature scheme and let I be the global information. The adversary A is given a random public key and access to the random hash oracle $\mathcal{H}(\cdot)$ and the signing oracle $\mathcal{S}_{I,sk}^{\mathcal{H}(\cdot)}(\cdot)$. The latter has access to the former oracle, takes a message as input and returns a signature of it computed using I, sk. The adversary can query its oracles and occasionally returns a pair (M, σ). The advantage of the adversary $\mathsf{Adv}^{sign}_{S,I}(A)$ is defined as the probability of A to output the valid message-signature pair, such that the message has not been queried to the signing oracle.

The signature scheme S is called to be *secure against existential forgery under chosen message attack* (or just a secure signature scheme) if there does not exist a polynomial-time[1] adversary A with non-negligible advantage $\mathsf{Adv}^{sign}_{S,I}(A)$.

3 The Gap-Diffie-Hellman (GDH) groups and the GDH signature scheme

We recall the basic signature scheme of [BLS]. It uses Gap-Diffie-Hellman groups, so accordingly we first provide the definitions for the latter.

3.1 GDH groups

Let G be a multiplicative group of the prime order p. We consider the following two problems in G.

Computational Diffie-Hellman (CDH) problem. Given (g, u, v), the three random elements of G, to compute $h = g^{\log_g u \cdot \log_g v}$.

Decisional Diffie-Hellman (DDH) problem. Given the four G elements (g, u, v, h), which with equal probability can be either all random elements of G or have the property that $\log_g u = \log_v h$, to output 0 in the former case and 1 otherwise.

We will refer to any four elements of G with the property defined above as a valid Diffie-Hellman (DH) tuple.

We now can define GDH groups. They are basically the groups where CDH problem is hard, while DDH problem is easy.

Definition 3.1 A prime order group G is a GDH group if there exists an efficient algorithm $\mathcal{V}_{\mathcal{DDH}}()$ which solves the Decisional Diffie-Hellman problem in G and there is no polynomial-time (in $|p|$) algorithm which solves the Computational Diffie-Hellman problem.

The details on the existence and composition of GDH groups see [BF, BLS, J, JN].

3.2 The GDH signature scheme

Let G be a GDH group. Let $[\{0,1\}^* \rightarrow G^*]$ be a hash function family, each member of which maps arbitrary long strings to G^* and H be a random member of this family. The global information I contains the generator g of G, p and a description of H. The algorithms $(\mathcal{K}, \mathcal{S}, \mathcal{V})$ of the GDH signature scheme $GS[G]$ are defined as follows.

- $\mathcal{K}(I)$: Parse I as (p, g, H). Pick random $x \stackrel{R}{\leftarrow} Z_p^*$ and compute $y \leftarrow g^x$. Return $(pk = (p, g, H, y), sk = x)$.
- $\mathcal{S}(I, sk, M)$: Parse I as (p, g, H). Compute $\sigma = H(M)^x$. Return (M, σ).
- $\mathcal{V}(pk, M, \sigma)$: Parse pk as (p, g, H, y). If $\mathcal{V}_{\mathcal{DDH}}(g, y, H(M), \sigma) = 1$ then return 1, else return 0.

In [BLS] the authors state and prove the following result.

Theorem 3.2 Let G be a GDH group. Then $GS[G]$ is a secure signature scheme in the random oracle model.

[1] Here and everywhere in the paper we assume that the complexity is measured as a function of a security parameter contained in the global info I. If I contains more than a security parameter we fix the randomized generator for I and the probability includes its choices.)

4 Robust proactive threshold GDH signature scheme

In this section we present a threshold version of GDH signature scheme which is robust, proactive and does not require a trusted dealer. The construction is very simple, since the structure of the base scheme permits to avoid many difficulties one needs to overcome while making threshold versions of many standard signature schemes, such as RSA, DSS, etc.

4.1 Definitions

We now recall the basic setting and notions of threshold signature schemes.

COMMUNICATION MODEL. As usual, the participants in our scheme are the set of n players $\{P_1, \dots, P_n\}$. All players are connected by a broadcast channel as well as by secure point-to-point channels.

THRESHOLD SECRET SHARING. The set of values $(s_1, \dots, s_n)$ is said to be a (t,n)-*threshold secret sharing* of the value s if any $k \leq t$ values from this set does not reveal any information about s and there exists an efficient algorithm which takes as input any $t+1$ values from this set and outputs s. We write $(s_1, \dots, s_n) \xrightarrow{(t,n)} s$.

THRESHOLD SIGNATURE SCHEMES AND THEIR SECURITY. Let $S = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ be a signature scheme and let I be the associated global information. A corresponding (t,n)-*threshold signature scheme* $TS = (\mathcal{TK}, \mathcal{TS}, \mathcal{V})$ consists of three algorithms, where the verification algorithm is the same as of S. A randomized distributed *threshold key generation* algorithm $\mathcal{TK}$ is an interactive protocol that takes I and is run by the players $P_1, \dots, P_n$. The protocol returns the public key pk, and the private output of each player P_i is a value x_i such that $(x_1, \dots, x_n) \xrightarrow{(t,n)} sk$, where sk is a secret key corresponding to pk. $\mathcal{TK}$ is said to *complete successfully* if it outputs (sk, pk) having the distribution the same as the output of $\mathcal{K}$. The distributed possibly randomized *threshold signature generation* algorithm $\mathcal{TS}$ is an interactive protocol run by the subset of the players, where the input of each player P_i is a message M, the global info I and the player's private input x_i. The algorithm can be considered as consisting of two interactive protocols: a *signature share generation* and *signature reconstruction*. At the end of the signature share generation protocol each player outputs its signature share. All signature shares are then combined using the signature reconstruction protocol. The output of the algorithm is a message-signature pair (M, σ). $\mathcal{TS}$ is said to *complete successfully* if it outputs (M, σ) such that $(M, \sigma) = \mathcal{S}(I, sk, M)$, for all $M \in \{0,1\}^*$.

Definition 4.1 Let I be the global info, $S = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ be a signature scheme and let $TS = (\mathcal{TK}, \mathcal{TS}, \mathcal{V})$ be the corresponding threshold signature scheme. TS is called secure robust threshold signature scheme if the following conditions hold:

1. *Unforgeability.* No polynomial-time adversary which is given I, is allowed to corrupt up to t players and given the view of the protocols $\mathcal{TK}, \mathcal{TS}$, the latter being run on the input messages of the adversary's choice, can produce the pair (M, σ) such that M has not been submitted by the adversary as public input to $\mathcal{TS}$.

2. *Robustness.* For every polynomial-time adversary that is allowed to corrupt up to t players, the protocols $\mathcal{TK}, \mathcal{TS}$ complete successfully.

In the above definition corruption means that an adversary chooses the players it wants to corrupt in advance and is allowed to alter the computation of the corrupted player in any way and to see their private inputs. If the above definition is adjusted to the random oracle model, then all the parties are given access to the random hash oracle.

4.2 TGS, the threshold GDH signature scheme

We now describe our construction.

Let G be a GDH group, $I = (p, g, H)$ be the global info and let $TGS[G] = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ be the GDH signature scheme as defined in Section 3.2. The algorithms $\mathcal{TK}, \mathcal{TS}$ of the corresponding threshold GDH signature scheme $TGS[G] = (\mathcal{TK}, \mathcal{TS}, \mathcal{V})$ are defined as follows.

- $\mathcal{TK}$ is exactly the distributed key generation protocol DKG for discrete-log based systems of Gennaro et al. [GJKR99][2]. It is jointly executed by a set of paries $\{P_1, \ldots, P_n\}$ It takes as input I and outputs a public key y. The private output of each player P_i is a share x_i such that $(x_1, \ldots, x_n) \xrightarrow{(t,n)} x$, where $x = \log_g y$. Any subset R of $t+1$ players can reconstruct x using well-known techniques of Lagrange interpolation: $x = \sum_{i \in R} L_i x_i$, where L_i is the appropriate Lagrange coefficient for the set R. As [GJKR99] shows, for each share x_i, the value $B_i = g^{x_i}$ can be computed from publicly available information broadcast during the protocol execution. Hence, we assume that these values are publicly available and will use them in order to achieve robustness.

- In order to execute the signature share generation protocol of $\mathcal{TS}$ each player P_i in the any subset of more than t players takes input a message M and its private share x_i and computes the share of the signature $\sigma_i = H(M)^{x_i}$ and broadcasts σ_i. The signature reconstruction protocol can be performed by any player or a set of players. We will assume for simplicity that it is run by some *designated* player D. In order to achieve robustness for D checks that $\mathcal{V}_{\mathcal{DDH}}(g, B_i, H(M), \sigma_i) = 1$each i. If this does not hold, new output requested from the corresponding player or it assumed malicious. Assuming wlog that R is a set of $t+1$ honest players, D computes the resulting signature $\sigma = \Pi_{i \in R}(\sigma_i^{L_i})$, where L_i is the appropriate publicly known Lagrange coefficient for the set R. The output of the protocol is (M, σ).

We summarized the properties of TGS in Section 1.

Theorem 4.2 Let G be a GDH group. Then $TGS[G]$ is a secure threshold signature scheme in the random oracle model against an adversary which is allowed to corrupt any $t < n/2$ players.

The proof of the above theorem is in Section A.

4.3 Adding proactive security

The idea of the proactive approach is to periodically renew shares of a secret such that information gained by an adversary learning some number of shares (less than a threshold) in one time period be useless for the adversary's next attacks in the future time periods when all shares are renewed. Proactive secret sharing algorithm PSS has been proposed in [HJKY]. In order to simplify an application of PSS [HJJKY] state the requirements on a threshold signature scheme for proactivization with the help of the PSS protocol. Namely, the authors prove that the security of the robust threshold signature scheme will be preserved when used with PSS protocol if it is a discrete-log based robust threshold signature scheme, which threshold key generation protocol implements

[2] We are interested in verifiable threshold key generation algorithms without a trusted dealer producing Shamir's secret sharing of a secret [S]. Some threshold signature scheme, e.g. threshold DSS proposed in [GJKR96] use the distributed key generation protocol (DKG) of Pedersen [P]. The intuition behind the latter protocol is to have n parallel executions of Feldman's verifiable secret sharing protocol [F], such that each player acts as a dealer. However, [GJKR99] point out the weakness of DKG of [P]. Namely, it is possible for a corruptive adversary to prevent the protocol by completing correctly by manipulating the distribution of the shared secret key. DKG protocol of [GJKR99] is based on the ideas similar to the protocol of [P], has comparable complexity, but provably fixes the weakness of the latter. We use the security results of [GJKR99] to prove the security of our threshold signature scheme.

Shamir's secret sharing of the secret signature key x corresponding to the public key $y = g^x$ and outputs verification information $(g^{x_1}, \ldots, g^{x_n})$, where $(x_1, \ldots, x_n)$ are secret shares of the players and if the threshold signature protocol is simulatable. As it is easy to note our threshold signature scheme $TGS = (\mathcal{TK}, \mathcal{TS}, \mathcal{V})$ meets all these requirements (recall that the verification information mentioned above is not explicitely output by $\mathcal{TK}$ but can be computed using publicly available information.) Thus TGS can be proactivized using PSS and methods of [HJKY, HJJKY]. We add that PSS outputs the verification information after each share update, hence the verification of signature shares can be conducted as before.

We now briefly summarize the properties of TGS. It is robust and can tolerate any $t < n/2$ malicious parties. Its key generation protocol does not require a trusted dealer. Its signature share generation protocol is basically the signing algorithm of the base scheme and the signature reconstruction requires only multiplication of shares. Therefore the signature generation protocol does not require interaction or any zero-knowledge proofs, and has a minimal overhead compared to that of the base scheme. The shares are short and their length does not depend on the number of parties. We also showed how proactive security can be added to our scheme using general methods of [HJKY, HJJKY].

We compared the new GDH threshold signature scheme with some other existing constructions in Section 1.

5 The GDH multisignature scheme

5.1 Multisignature schemes

Let $P = \{P_1, \ldots, P_n\}$ be a group of n players. Let I be the global information string. The algorithms of a multisignature scheme $MS = (\mathcal{MK}, \mathcal{MS}, \mathcal{MV})$ are defined as follows. A randomized *key generation* algorithm $\mathcal{MK}$ takes a global information I and outputs a pair (sk, pk) of a secret and a public keys. Each player $P_i \in P$ runs this algorithm and as a result obtains a pair of secret and public keys (sk_i, pk_i). A possibly randomized *multisignature generation* algorithm $\mathcal{MS}$ is an interactive protocol run by an arbitrary subset of players $L \subseteq P$. The input of each player $P_i \in L$ is a message $M \in \{0,1\}^*$, the global info I and the player's secret key sk_i. The output of the algorithm is a triple $T = (M, L, \sigma)$ consisting of the message, description of the subgroup L and the multisignature. A deterministic *verification* algorithm $\mathcal{MV}$ takes the public keys of all players in L and T and outputs 1 (accepts) or 0 (rejects).

Note that it is up to a particular application to decide what subgroup is required to sign a message. A person who verifies the validity of a multisignature might reject it not because it's invalid but because she is not satisfied with the subgroup which signed the message. We leave it to applications to agree each time on the desired subgroup of signers and for the analysis we do not take this problem into account.

5.2 MGS, the GDH multisignature scheme

We now describe the new multisignature scheme $MGS = (\mathcal{MK}, \mathcal{MS}, \mathcal{MV})$ which is based on the GS signature scheme of [BLS] we recalled in Section 3.2. The construction is very simple and efficient, and it also solves an open problem stated in [MOR], namely, to find a provably secure multisignature scheme where the composition of the subgroup can be decided after the signature shares are computed by the signers.

Let G be a GDH group and let I be the global information that consists of a generator g of G, $p = |G|$ and a description H of a random member of the family of hash functions $[\{0,1\}^* \rightarrow G^*]$. Let

$P = \{P_1, \ldots, \mathrm{P}_n\}$ be the group of players.

- $\mathcal{MK}$: The same as the key generation algorithm $\mathcal{K}$ of GDH signature scheme GS (see Section 3.2.) Recall that each player P_i runs this protocol and obtains public key $pk_i = (p, g, H, y_i)$ and secret key $sk_i = x_i$, where $y_i = g^{x_i}$.

- $\mathcal{MS}$: Any player $P_j \in P$ with a secret key $sk_j = x_j$, that wishes to participate in signing takes M, computes and broadcasts $\sigma_j \leftarrow H(M)^{x_j}$. Let $L = \{P_{i_1}, \ldots, P_{i_l}\}$ be a subgroup of players contributed to the signing. Let $J = \{i_1, \ldots, i_l\}$ denote the set of indeces of such players. The designated signer D (which can be implemented by any player) that we assume wlog knows the signer of each signature computes $\sigma = \Pi_{j \in J}(\sigma_j)$ and outputs $T = (M, L, \sigma)$.

- $\mathcal{MV}$: The verifier takes $T = (M, L, \sigma)$ and the list of public keys of the players in L: $(pk_{i_1}, \ldots, pk_{i_l})$, where $pk_{i_j} = g^{x_{i_j}}$ for each $i_j \in J$. The verifier computes $pk_L = \Pi_{j \in J}(pk_j) = \Pi_{j \in J}(g^{x_j})$ and outputs $\mathcal{V}_{\mathcal{DDH}}(g, pk_L, H(M), \sigma)$.

The robustness property can be added to MGS if D verifies the validity of each signature it receives. We provided the comparison of MGS with other multisignature schemes in Section 1.

Batch verification of GS signatures. The approach underlying the above multisignature scheme can easily be applied to provide efficient batch verification of several GS signatures of the same message under different public keys[3]. A verifier needs first to play the role of D above to multiply the given signatures and then continue the verification according to the verification algorithm above.

5.3 Security of multisignatures

The notion of security for multisignatures has to capture the possibility of an adversary to "forge" a subgroup L and a multisignature of some message such that the latter is accepted by a verifier when not all players of the subgroup L did sign the message. In other words, no valid multisignature should keep an honest player that part of L accountable if it did not participate in signing.

In order to achieve its goal an adversary might corrupt players, send arbitrary messages during multisignature generation protocol, etc. We also allow an adversary to create arbitrary keys for corrupted players, possibly dependent on the keys of honest players, in order to model well-known rogue-key attacks. With respect to these attacks we put only one limitation on the adversary, namely we require it to prove knowledge of secret keys during the public key registration, which is (or should be) the standard practice. We model this for simplicity by asking the adversary to output public and secret key of corrupted users in key generation algorithm. Alternatively we could ask the adversary to provide proofs of knowledge so we be able to extract secret keys, however, this would unnecessary complicate the model. We allow an adversary to corrupt all but one player and its goal is to "frame" the honest player.

We note that such a powerful adversary can always deviate from the protocol thus preventing generating a valid multisignature. Similarly to [MOR] we do not focus on the robustness property in this work. We will sketch, however, how our multisignature scheme can be made robust.

We now formalize the notion of security for multisignatures. It is similar to the one given in [MOR], however, our definition is more general in that an idividual signer does not have to know the subgroup of co-signers.

[3]This problem is orthogonal to the problem of batch verification of signatures of the different messages under the same key, which has been addressed in [BGR].

Definition 5.1 An adversary A learns the global info I and a randomly generated public key pk_1 corresponding to a single honest player. Wlog we refer to the honest player P_1. A generates and outputs the rest of $n-1$ pairs of public and secret keys and is allowed to run multisignature generation protocol with the honest player on behalf of $n-1$ corrupted players on the messages chosen by the adversary. The advantage of the adversary $\mathsf{Adv}^{mult}_{MS,I}(A)$ is defined as the probability of A to output the valid message–subgroup-signature triple (M, L, σ), such that $P_1 \in L$, $\mathcal{MV}(M, L, \sigma) = 1$ and P_1 did not complete the multisignature generation protocol on the input message M.

We say that a multisignature scheme MS is secure *against existential forgery under chosen message attack* (or just secure multisignature scheme) if there does not exist a polynomial-time adversary A with non-negligible advantage $\mathsf{Adv}^{mult}_{MS,I}(A)$. ▮

As usual, in order to adjust the above definition to the random oracle model all parties and the signing oracle are given access to the random hash oracle.

5.4 Security of the MGS multisignature scheme

Theorem 5.2 Let G be a GDH group. Then $MGS[G]$ is a secure multisignature scheme in the random oracle model.

The proof of the above theorem can be found in Section B.

6 The blind GDH signature scheme

The syntax of the key generation and verification algorithms $\mathcal{BK}, \mathcal{BV}$ of a blind signature scheme $BS = (\mathcal{BK}, \mathcal{BS}, \mathcal{BV})$ is the same as the one of the corresponding algorithms of a regular signature scheme. The *blind signing* algorithm $\mathcal{BS}$ is an interactive protocol between a *user* and a *signer*, where the former knows the public key and the latter is given the global info and the secret key. And the end of the protocol the user outputs a message-signature pair (M, σ). It is required that if (M, σ) is the output of the blind signing algorithm, then $\mathcal{V}(pk, M, \sigma) = 1$.

6.1 BGS, the blind GDH signature

We now propose a new blind signature scheme based on GDH signature scheme.

Let G be a GDH group. Let $I = (p, g, H)$ be the global info. Let $GS[G]$ be the GDH signature scheme of [BLS] we recalled in Section 3.2. The blind GDH signature scheme $BGS[G] = (\mathcal{BK}, \mathcal{BS}, \mathcal{BV})$ is defined as follows. The algorithms $\mathcal{BK}, \mathcal{BV}$ are the same as those of GS. The blind signing protocol $\mathcal{BS}$ is defined as follows. The user holds a public key $pk = (p, g, H, y)$. In order to "blindly" sign a message $M \in \{0,1\}^*$ the user picks a random number $r \stackrel{R}{\leftarrow} Z_p^*$, computes $\overline{M} = H(M) \cdot g^r$ and sends it to the signer. The signer knows $I = (p, g, H), sk = x$. The signer computes $\overline{\sigma} = (\overline{M})^x$ and sends it to the user. The latter computes $\sigma \leftarrow \overline{\sigma} \cdot y^{-r}$ and outputs (M, σ).

Note that above $\sigma = H(M)^x$, that is a valid signature on M.

6.2 Security of blind signatures

The notion of security of blind signatures captures two properties. The first property is "*blindness*", meaning the signer in the blind signing protocol should not learn any information about the messages the user obtained signatures on. The second property is a special form of *unforgeability*, namely, the user that has been engaged in l runs of the blind signing protocol should not be able to obtain more than l signatures. The standard notion of unforgeability under chosen-message attack of digital signatures [GMR] cannot be used as a notion of unforgeability for blind signatures since by their construction a user has to be able to produce a valid signature of a previously unsigned message. The accepted formalization of security for blind signature is security *against one-more-forgery* [PS, PS1].

Definition 6.1 Let $S = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ be a signature scheme and let $BS = (\mathcal{BK}, \mathcal{BS}, \mathcal{BV})$ be the corresponding blind signature scheme. An adversary A learns the public key pk randomly generated by $\mathcal{BK}$. A is allowed to play the role of a user in the runs of the blind signing protocol. After interactions with the signer A outputs some number of message-signature pairs. The advantage of the adversary $\mathsf{Adv}^{blind}_{BS,I}(A)$ is defined as the probability of A to output a set L of valid message-signature pairs, such that the number of invoked blind signing protocols with the signer is strictly less than the size of L.

We say that the blind signature scheme BS is secure *against one-more forgery under chosen message attack* or just secure blind signature scheme if there does not exist a polynomial-time adversary A with non-negligible advantage $\mathsf{Adv}^{blind}_{BS,I}(A)$.

6.3 Chosen target CDH assumption

First we claim that BGS has the blindness property. This is because the signer receives only random elements in G which are independent of the outputs of the user.

Similarly to the proof of security of the Chaum's RSA-based blind signature scheme [C] given in [BNPS] we reduce the security in the sense of unforgeability of the blind signature scheme to the *chosen-target* version of the appropriate computational assumption. Security of the RSA blind signature is proven secure assuming hardness of the chosen-target RSA inversion problem [BNPS]. Namely, the assumption states that for a randomly generated RSA key pair $pk = (N, e), sk = (N, d)$[4] no polynomial time adversary which is given pk, the "target" oracle which outputs random target points in Z^*_N and the "helper" RSA inversion oracle $(\cdot)^d \bmod N$ can invert (compute $(\cdot)^d \bmod N$) any subset of the target points such that the number of queries to the helper RSA inversion oracle is strictly less than the number of queries to the target oracle. It is suggested in [BNPS] that an analogue of this assumption can be formulated for any family of one-way functions. We propose the following analogous problem and the assumption.

Definition 6.2 [The chosen-target CDH problem and assumption] Let $G =< g >$ be a group of a prime order p. Let x be a random element of Z^*_p and let $y = g^x$. Let H be a random instance of a hash function family $[\{0,1\}^* \to G^*]$ The adversary B is given (p, g, H, y) and has access to the target oracle $\mathcal{T}_G$ that returns random points z_i in G and the helper oracle $(\cdot)^x$. Let q_t, (resp. q_h) be the number of queries B made to the target (resp. helper) oracles. The advantage of the adversary attacking the chosen-target CDH problem $\mathsf{Adv}^{ct-cdh}_G(B)$ is defined as the probability of B to output a set V of, say, l pairs $((v_1, j_1), \ldots (v_l, j_l))$, where for all $1 \le i \le l \; \exists \; 1 \le j_i \le q_t$ such that $v_i = z^x_{j_i}$, all v_i are distinct and $q_h < q_t$.

The chosen-target CDH assumption states that there is no polynomial-time adversary B with non-negligible $\mathsf{Adv}^{ct-cdh}_G(B)$. ▮

Note that if the above adversary makes one query to the target oracle then the chosen-target CDH assumption is equivalent to the standard CDH assumption. We assume that the chosen-target CDH problem is hard for all groups where CDH problem is hard; this includes GDH groups.

[4]Here $N = pq$ is a product of two random primes, e is a random element of $Z^*_{\phi(N)}$ and $ed \equiv 1 \bmod \phi(N)$, where $\phi(\cdot)$ is a Euler's totient function.

6.4 Security of the BGS blind signature scheme

Theorem 6.3 If the chosen-target CDH assumption is true in the group G then the $BGS[G]$ blind signature scheme is secure against one-more forgery under chosen message attack.

We provide the proof in Section C.

7 Acknowledgements

We thank Mihir Bellare for the useful discussions, for suggesting to consider the topic of threshold signature schemes and for convincing us to finally put the ideas into writing. We thank Daniele Micciancio and Adriana Palacio for their useful comments on the draft of this paper. We also thank Leonid Reyzin for clarifications on [MOR].

References

[BR] M. Bellare and P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols. *First ACM Conference on Computer and Communications Security*, ACM, 1993.

[BNPS] M. Bellare, C. Namprempre, D. Pointcheval and M. Semanko, "The One-More-RSA-Inversion Problems and the security of Chaum's Blind Signature Scheme," *e-print archive, report #2001/02*. The preliminary version of this paper "The Power of RSA Inversion Oracles and the Security of Chaum's RSA-Based Blind Signature Scheme" *appeared in Financial Cryptography 01, Lecture Notes in Computer Science* Vol. 2339, P. Syverson ed, Springer-Verlag, 2001.

[BGR] M. Bellare, J. Garay and T. Rabin, "Fast batch verification for modular exponentiation and digital signatures ," *Advances in Cryptology – Eurocrypt '98*, LNCS Vol. 1403, K. Nyberg ed., Springer-Verlag, 1998.

[BW] E. Berlekamp and L. Welch, "Error correction of algebraic block codes," *US Patent 4,633,470.*

[BF] D. Boneh and M. Franklin. "Identity-based encryption from the Weil Pairing," *Advances in Cryptology – Crypto '01*, LNCS Vol. 2139, J. Kilian ed., Springer-Verlag, 2001.

[BLS] D. Boneh and B. Lynn and H. Shacham, "Short signatures from the Weil pairing," *Advances in Cryptology – ASIACRYPT '01*, LNCS Vol. 2248, C. Boyd ed., Springer-Verlag, 2001.

[B] C. Boyd, "Digital multisignatures," *Cryptography and Coding*, 1986

[CS] J. Camenisch and M. Stadler, "Efficient group signatures for large groups," *Advances in Cryptology – Crypto '97*, LNCS Vol. 1294, B. Kaliski ed., Springer-Verlag, 1997.

[C] D. Chaum, "Blind signatures for untreaceable payments,"*Advances in Cryptology – Crypto '82*, LNCS Vol. , ed., Springer-Verlag, 1982.

[CvH] D. Chaum and E. van Heyst, "Group signatures," *Advances in Cryptology – Eurocrypt '91*, LNCS Vol. 547, D. Davies ed., Springer-Verlag, 1991.

[CH] R. Canetti and A. Herzberg, " Maintaining security in the presence of transient faults," *Advances in Cryptology – Crypto '94*, LNCS Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.

[D88] Y. Desmedt, "Society and group oriented cryptography,"*Advances in Cryptology – Crypto '87*, LNCS Vol. 293, C. Pomerance ed., Springer-Verlag, 1987.

[D94] Y. Desmedt, "Threshold cryptography,", *European Transactions on Telecommunications*, 5(4), 1994.

[DF89] Y. Desmedt and Y. Frankel, "Threshold cryptosystems," *Advances in Cryptology – Crypto '89*, LNCS Vol. 435, G. Brassard ed., Springer-Verlag, 1989.

[DF91] Y. Desmedt and Y. Frankel, "Shared generation of authenticators and signatures," *Advances in Cryptology – Crypto '91*, LNCS Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.

[F] P. Feldman "A practical scheme for non-interactive verifiable secret sharing," Proceedings *of the* 28th *Symposium on Foundations of Computer Science*, IEEE, 1987.

[FD] Y. Frankel and Y. Desmedt "Parallel reliable threshold multisignature," *Technical report TR-92-04-02, University of Wisconsin-Milwaukee*, 1992.

[FGMY] Y. Frankel, P. Gemmal, P. MacKenzie and M. Yung, "Proactive RSA," *Advances in Cryptology – Crypto '97*, LNCS Vol. 1294, B. Kaliski ed., Springer-Verlag, 1997.

[GMLS] S. Galbraith, J. Malone-Lee, N. P. Smart, "Public key signatures in the multi-user setting", *Information Processing Letters*, Vol. 83, Issue 5, 2002.

[GJKR96] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "Robust threshold DSS signatures," *Advances in Cryptology – Eurocrypt '96*, LNCS Vol. 1070, U. Maurer ed., Springer-Verlag, 1996.

[GJKR99] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems", *Advances in Cryptology – Eurocrypt '99*, LNCS Vol. 1592, J. Stern ed., Springer-Verlag, 1999.

[GMR] S. Goldwasser, S. Micali and R. Rivest, "A digital signature secure against adaptive chosen-message attacks", *SIAM Journal on Computing*, 17(2):281-308, 1988.

[H] L. Harn, "Group-oriented (t,n) threshold digital signature scheme and digital multisignature," *IEE Proc. Computers and Digital Techniques*, 141(5), 1994.

[HJJKY] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk and M. Yung, "Proactive public key and signature systems," *ACM Conference on Computers and Communication Security*, 1997. 22:612-613, (1979).

[HJKY] A. Herzberg, S. Jarecki, H. Krawczyk and M. Yung, "Proactive secret sharing, or: How to cope with perpetual leakage,", *Advances in Cryptology – Crypto '95*, LNCS Vol. 963, D. Coppersmith ed., Springer-Verlag, 1995.

[HMP] P. Horster, M. Michels and H. Petersen, "Meta-multisignatures schemes based on the discrete logarithm problem," *IFIP/Sec* 1995.

[IN] K. Itakura and K. Nakamura, "A public key cryptosystem suitable for digital multisignatures," *NEC Research & Development*, 71:1-8, 1983.

[J] A. Joux, "A one-round protocol for tripartite Diffie-Hellman," *ANTS-IV conference, vol. 1838*.

[JN] A. Joux and K. Nguyen, "Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups," *e-print archive, report #2001/03*.

[LHL] C. Li, T. Hwang and N. Lee, "Threshold-multisignature schemes where suspected forgery implies traceability of adversarial shareholders," *Advances in Cryptology – Eurocrypt '94*, LNCS Vol. 950, A. De Santis ed., Springer-Verlag, 1994.

[L] A. Lysyanskaya, "Unique signatures and verifiable random functions from the DH-DDH separation", *Advances in Cryptology, Crypto'02*, 2002.

[MOR] S. Micali, K. Ohta and L. Reyzin, "Accountable-subgroup multisignatures," *ACM Conference on Computer and Communications Security*, 2001.

[O] T. Okamoto, "A digital multisignature schema using bijective public-key cryptosystems," *ACM Transaction on Computer Systems*, 6(4): 432-441, 1988.

[OO91] K. Ohta and T. Okamoto, "A digital multisignature scheme based on the Fiat-Shamir scheme", *Advances in Cryptology – ASIACRYPT '91*, LNCS Vol. 739, H. Imai, R. Rivest and T. Matsumoto ed., Springer-Verlag, 1991.

[OO99] K. Ohta and T. Okamoto, "Multi-signature scheme secure against active insider attacks", *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E82-A(1):21-31, 1999.

[OY] R. Ostrovsky and M. Yung, "How to withstand mobile virus attacks," *PODC*, 1991.

[P] T. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," *Advances in Cryptology – Eurocrypt '91*, LNCS Vol. 547, D. Davies ed., Springer-Verlag, 1991.

[PS] D. Pointcheval and J. Stern, "Provably secure blind signature schemes,"*Advances in Cryptology – ASIACRYPT '96*, LNCS Vol. 1163, ?? ed., Springer-Verlag, 1996.

[PS1] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of cryptology,* 13(3):361-396, 2000.

[R] T. Rabin, "A simplified approach to threshold and proactive RSA," *Advances in Cryptology – Crypto '98*, LNCS Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.

[RST] R. Rivest, A. Shamir and Y. Tauman, "How to leak a secret", *Advances in Cryptology – ASIACRYPT '01*, LNCS Vol. 2248, C. Boyd ed., Springer-Verlag, 2001.

[S] A. Shamir, "How to share a secret," *Communications of the ACM,* 22:612-613, (1979).

[Sh] V. Shoup, "Practical threshold signatures", *Advances in Cryptology – Eurocrypt '00*, LNCS Vol. 1807, B. Preneel ed., Springer-Verlag, 2000

A Proof of Theorem 4.2

Proof: We start from the robustness property. The correctness properties of DKG proven in [GJKR99] are: in the presence of a an adversary that corrupts $t < n/2$ players all subsets of $t+1$ shares define the same unique x that correspond to the unique public key $y = g^x \bmod p$, x is uniformly distributed in Z_p^* and thus y is uniformly distributed in $G =< g >$. This implies that $\mathcal{TK}$ that is DKG completes successfully even in the presence of a corruptive adversary. Now note that only valid signature shares can pass the verification step of $\mathcal{TS}$ since each valid signature share σ_i is a valid GS signature corresponding to the public key $g^{x_i} = B_i$ and the verification step of $\mathcal{TS}$ is analogous to the signature verification algorithm $\mathcal{V}$ of $GS[G]$. Now assuming the resulting signature σ is computed as multiplication of $t+1$ valid shares of the form $H(M)^{L_i \cdot x_i}$ due to Lagrange interpolation $\sigma = H(M)^x$ which means that $\mathcal{TS}$ complete successfully.

We now prove the unforgeability property. As shown in [GJKR96] the following implies unforgeability of $TS = (\mathcal{TK}, \mathcal{TS}, \mathcal{V})$: if the underlying signature scheme $S = (\mathcal{K}, \mathcal{S}, \mathcal{V})$ is secure and if TS is simulatable, where the latter condition means that for every probabilistic polynomial-time adversary A that corrupts up to t players there exists a probabilistic polynomial-time simulator SIM that on input the public key y and M the signature σ of M can simulate the view for A that is polynomially indistinguishable from A's view of a run of $\mathcal{TK}$ protocol that outputs y and the following run of $\mathcal{TS}$ that outputs σ. The simulator for a view of a run of $\mathcal{TK}$ of an adversary which corrupts any $t < n/2$ players is presented in [GJKR99] as their proof of security of DKG. Wlog [GJKR99] assumes that A corrupts the players with indices $1, \ldots, t'$ where $t' \leq t$. So will do we in our discussion. In particular, it is shown in [GJKR99] that SIM knows all the shares x_i but one (wlog assumed to be x_n) of the honest players, and the values corresponding to the last share can be computed using the fixed y and the rest of the shares. All the shares have the right distribution (note that the last share is not known to SIM, but it is implicitly used in the values that SIM computes and outputs.) We use a similar approach in order to extend the construction of SIM to simulate the view of A in the run of $\mathcal{TS}$. SIM can verify the validity of signature shares output by corrupted parties being honest during the run of $\mathcal{TK}$ using knowledge of their shares. It needs to simulate the signature shares of the uncorrupted parties. Since SIM has all the shares $x_{t'+1}, \ldots, x_{n-1}$ it can create signature shares as $\sigma_i = H(M)^{x_i}$ for $i = t'+1, \ldots, n-1$. SIM creates the signature share corresponding to the player P_n as $\sigma_n = \sigma \cdot \Pi_{i \in R}(\sigma_i)^{-1}$. Since as shown [GJKR99] all the shares used by SIM have the right distribution (including the one corresponding to the n's player which SIM implicitly does not know,) then all the signature shares computed by SIM have the right distribution. This is because all signature shares but σ_n explicitely use known corresponding secret shares and by construction σ_n corresponds to the share x_n implicitly used by SIM.

The above arguments together with Theorem 3.2 imply the statement of the theorem. Note that random oracle assumption is used only by the proof of security of the base scheme. ∎

B Proof of Theorem 5.2

Proof: Let A be a poly-time adversary for the GDH multisignature scheme $MGS[G]$. We will construct the adversary B for the $GS[G]$ signature scheme such that $\mathsf{Adv}^{mult}_{MGS[G],I}(A) = \mathsf{Adv}^{sign}_{GS[G],I}(B)$. This means that if $GS[G]$ is a secure signature scheme then $MGS[G]$ is a secure multisignature scheme. Having the result of Theorem 3.2, the statement of the above theorem follows.

The intuition of the proof is that if A manages to frame an honest player by constructing a valid multisignature on some message without interaction with this honest player, then B can derive a forgery on the previously unsigned message. We now give the details. B is having a public key y and access to the random hash oracle and the signing oracle. B will run A simulating for it the single honest player. First B gives A the public key $pk_1 = y$. Then A outputs the set of $n-1$ public and secret key pairs $(y_2, x_2), \ldots, (y_n, x_n)$, where $y_i = g^{x_i}$ for $1 < i \leq n$. B simulates A's random hash oracle using its own oracle. Whenever A asks the honest player to participate in multisignature generation protocol on some message M, B forwards the query to its signing oracle and returns the reply back to A. At some point A outputs an attempted forgery $T = (M, L, \sigma_L)$. Wlog suppose $y_{j_1}, \ldots, y_{j_l}$ are the public keys of the players that constitute L. Then B computes $\sigma = \sigma_L \cdot \Pi_{j \in J/\{1\}}(H(M)^{-x_j})$ and outputs (M, σ). It is easy to see that B simulates for A a valid experiment, runs in time comparable to running time of A and that it succeeds in forgery whenever A is successful (in this case $\sigma = H(M)^x$.) ▮

C Proof of Theorem 6.3

Proof: Let G be a GDH group and let $I = (p, g, H)$ be the global info. Let A be any poly-time adversary attacking $BGS[G]$ against one-more forgery under chosen message attack. We will present a poly-time adversary B for the chosen-target CDH problem such that $\mathsf{Adv}^{blind}_{BS[G],I}(A) = \mathsf{Adv}^{ct-cdh}_{G}(B)$. The statement of the theorem follows.

We note that since the signer in $\mathcal{BS}$ protocol of BGS scheme has only one move, it is enough in the definition of security of Definition 6.1 to give A access to a *blind signing oracle* $(\cdot)^x$, where x is a secret input hold by the signer. Since we analyze security of BGS in the random oracle model, the adversary A is also given access to the random hash oracle $\mathcal{H}(\cdot)$.

We now describe the algorithm of B which will simulate A in order to solve the chosen-target CDH problem. The adversary B is given (p, g, H, y) and the target and helper oracles. B first provides A with the public key $pk = (p, g, H, y)$. B has to simulate the random hash oracle and the blind signing oracle for A. Each time A makes a new hash oracle query, that is distinct from the previous hash queries, B forwards it to its target oracle, returns the reply to A and adds this query and the reply to the stored list of such pairs. If A makes a hash query that it already made before, B replies consistently with an old reply. When A makes a query to the blind signing oracle, B re-sends it to its helper oracle $(\cdot)^x$ and forwards the answer to A. At some point A outputs a list of message-signature pairs $((M_1, \sigma_1), \ldots, (M_l, \sigma_l))$. For each $1 \leq i \leq l$ B finds M_i in the list of stored hash oracle queries and replies. Let j_i be the index of the found pair. B returns the list $(\sigma_1, j_1), \ldots, (\sigma_l, j_l)$ as its own output.

It is easy to see that the view of A in the simulated experiment is indistinguishable from its view in the real experiment and that B is successful only if A is successful. Thus $\mathsf{Adv}^{blind}_{BS,I}(A) = \mathsf{Adv}^{ct-cdh}_{G}(B)$ ▮