Expander Graphs for Digital Stream Authentication and Robust Overlay Networks

Dawn Song* David Zuckerman† J. D. Tygar*

*UC Berkeley, †UT Austin

{dawnsong,tygar@cs.berkeley.edu, diz@cs.utexas.edu}

Abstract

We use expander graphs to provide efficient new constructions for two security applications: authentication of long digital streams over lossy networks and building scalable, robust overlay networks. Here is a summary of our contributions: (1) To authenticate long digital streams over lossy networks, we provide a construction with a provable lower bound on the ability to authenticate a packet — and that lower bound is independent of the size of the graph. To achieve this, we present an authentication expander graph with constant degree. (Previous work, such as [MS01], used authentication graphs but required graphs with degree linear in the number of vertices.) (2) To build efficient, robust, and scalable overlay networks, we provide a construction using undirected expander graphs with a provable lower bound on the ability of a broadcast message to successfully reach any receiver. This also gives us a new, more efficient solution to the decentralized certificate revocation problem [WLM00].

1 Introduction

In this paper, we explore techniques for increasing the security and reliability of digitally broadcast data over networks. We explore two problems: proving the identity of a source of digitally broadcast data (authenticating the data stream) and constructing highly robust overlay networks.

1.1 Authenticating Digital Broadcast Data

On the Internet, digital broadcast and multicast are arguably among the most exciting and important mechanisms for communication. They allow a broad variety of content to reach a mass audience, and we now expect that a variety of digital content, including digital radio, digital television, and digital news will be available as digital streams at our office and home computers. In most settings, a single sender broadcasts a stream of data to a set of intended receivers. If we are concerned about the security of the broadcast data, we will want to ensure that the digitally broadcast streams truly originate from the purported source. In some ways, this problem is analogous to determining the source of a single point-to-point message, and we say we want to *authenticate* the data stream. The challenge is compounded because the Internet (and other networks) are not perfect: the networks often lose packets, and because of the nature of broadcast applications, those lost packets are generally not retransmitted. How can we efficiently authenticate broadcast data streams over lossy networks? In this paper, we present a powerful new construction based on expander graphs. We show that using this scheme we can prove a lower bound on the probability that a packet can be authenticated, and that this lower bound is *independent* of the size of the expander graph.

*This research was suported in part by the Defense Advanced Research Projects Agency under DARPA contract N6601-99-28913 (under supervision of the Space and Naval Warfare Systems Center San Diego), by the National Science foundation under grant FD99-79852, and by the United States Postal Service under grant USPS 1025 92-01-Z-0236. Views and conclusions contained in this document are those of the authors and do not necessarily represent the official opinion or policies, either expressed or implied of the US government or any of its agencies, DARPA, NSF, or USPS.

†Some of this work was done while the author was on leave at the University of California at Berkeley. Supported in part by a David and Lucile Packard Fellowship for Science and Engineering and NSF Grant CCR-9912428.

To understand the significance of this result, it is useful to review some alternative approaches to authenticating digital broadcast streams. A first naive approach to this problem is to use a secret shared between both the sender and all receivers. The sender uses this shared key to compute, for each packet, a message authentication code (MAC), and appends that MAC to the packet. Each receiver uses the shared secret key to verify the MAC. However, this approach has clear problems: since all receivers know the shared secret, any of them could forge or leak the shared secret. Anyone with the shared secret could successfully forge streams of broadcast data with correct MACs. A simple MAC by itself can not provide source authentication unless all receivers are trusted.

A second naive approach to the problem of authenticating digital broadcast streams is to have the sender use asymmetric cryptography. The sender could sign each packet with its private key, and each receiver could verify the signature of each packet with the corresponding public key. This does provide source authentication of each packet, but has heavy overhead for both generation and verification of packets. One might attempt to amortize this cost by computing a single digital signature over a large number of packets, but since the network is lossy, this poses problems for verification: how can a receiver verify a digital signature over a message with missing packets? Simple digital signatures can not provide efficient source authentication when the network is lossy.

In a sequence of important papers, several researchers have proposed a powerful new approach, *graph-based authentication* [GR97, WL98, PCTS00, GM01, MS01]. In graph-based authentication we amortize the cost of authentication over a stream. We sign a small set of packets, called *signature packets*. We view the digital stream as a directed acyclic graph, and each vertex represents a packet. If vertex i has an outgoing edge to vertex j, then packet i should contain the hash value of packet j. (We assume that our hash functions are collision resistant.) If we can successfully authenticate the signature packets, and if we can find a path in the graph from a signature packet vertex to a vertex k, then packet k is authenticated. The research questions include: (1) how to yield low communication overhead when the authentication graphs are converted to streams; (2) how to make the probability high of being able to authenticate a packet (i.e., find a path going from a signature packet vertex to a particular vertex) in the face of packet loss.

In this paper, we use expander graphs to construct an authentication protocol. Unlike previous work, our graphs have constant degree (previous work [MS01] required the graphs have degree linear in the number of vertices in the graph). Since the degree of the graph corresponds to the number of hashes that must be included in packets — this says something about the efficiency of our protocol: we only require a constant number of hashes to be included in each packet. Moreover, we also prove a lower bound on the probability that a packet can be authenticated upon arrival and our lower bound is independent of the size of the graph.

1.2 Robust Overlay Networks

Overlay networks, such as the MBone [Eri94], ABone [ABo], 6Bone [6Bo], and Gnutella [Gnu], allow system designers to build new distributed applications and protocols on top of the Internet. An overlay network is formed from a subset of nodes in the underlying network. Participating nodes communicate via *virtual links* between two nodes that may not be directly connected in the underlying network. A single virtual link may correspond to a network path that consists of a single hop, multiple hops, or even a dynamically changing underlying routing. These virtual links form the topology of the overlay network. By considering the problem in virtual links, we can abstract away from the Internet's basic structure, allowing us to rapidly build a variety of innovative applications. Overlay networks are particularly appropriate for highly decentralized applications, such as peer-to-peer file sharing. Because the applications are decentralized, overlay networks often change rapidly. In this paper, we model overlay networks by assuming that virtual links are reliable and do not fail, but that nodes may join or leave the network dynamically.

Consider the broadcast in an overlay network. For example, Gnutella, a popular file sharing service, broadcasts queries for particular files to neighboring nodes in the overlay network. These queries continue to propagate to other nodes. How can we provide efficient, robust broadcast in this model? Again, the problem can be viewed as a graph theoretic question: nodes in the network correspond to vertices in a graph, and virtual links in the network correspond to (undirected) edges in a graph. We construct an overlay network with efficient and robust broadcast using undirected expander graphs. This overlay network has constant degree, and we can prove a lower bound on the reachability of any node in the network. Furthermore, the maximum length from a given node to reach any node in the network is $O(\log n)$ virtual links. Since each node in the overlay network has a small virtual degree, and since between any two nodes in the overlay network there is a short virtual path, our construction allows particularly efficient realizations of overlay networks.

Our construction of efficient, robust overlay networks has an additional consequence: it gives us a new, decentralized way to distribute revocation lists (such as public key certificate revocation lists). The previous best approach was due to Wright, Lincoln, and Millen [WLM00]: In their model, each certificate has a list of *dependers*, i.e. hosts who are potentially interested in the revocation of the certificate. These dependers form a graph rooted at the owner of the certificate. When a certificate needs to be revoked, the owner of the certificate will broadcast a revocation request to all nodes that propagates via edges (links) in the graph. (This model is particularly valuable when the certificates are issued by individuals such as in PGP [Zim95].) Any node in the overlay network may be up or down, and the goal is to distribute revocations quickly to all up nodes. Wright et al. proposed a depender graph construction where the degree is a constant k and can tolerate at most $k-1$ node failures. If the probability of individual node failure remains constant, as the number of nodes in a network grows, k must grow linearly, so the Wright et al. approach does not scale to large networks. However, if we use our overlay networks as depender graphs, we get a scheme where: (1) we can prove a lower bound on the probability of each node receiving a revocation list; (2) the degree of each vertex (node in the network) is constant regardless of the size of the network; and (3) the maximum number of links from any node to any other node is $O(\log n)$.

1.3 Organization of This Paper

The rest of the paper is organized as follows. We review graph-based authentication of digital streams and expander graphs in Section 2. We describe our new construction of expander-based authentication for digital streams and provide the analysis in Section 3. In Section 4, we show how to apply our analysis technique to undirected expander graphs and construct new overlay networks using undirected expander graphs. We also show how to use our construction of overlay networks to provide a more efficient solution to the decentralized certification revocation problem. We review related work and discuss other issues in Section 5, and conclude in Section 6.

2 Preliminaries

2.1 Graph Based Authentication

Consider a sender transmitting consecutive packets $\{P_0, \ldots, P_{n-1}\}$ in a broadcast data stream. We construct an *authentication graph* to authenticate received packets. In particular, we construct a directed acyclic graph of n vertices where a vertex i corresponds to the packet P_i. Let (i, j) denote a directed edge starting from i and ending at j. An edge (i, j) in the graph indicates the authentication relationship between packet P_i and P_j: upon receiving packet P_i and P_j, if a receiver can authenticate both the contents and the source of P_i, then it can authenticate the contents and the source of P_j. We achieve this relationship by embedding the hash value of packet P_j into packet P_i. We assume the hash function is collision resistant, i.e., it is computationally infeasible to find two different values that hash to the same value. In practice, we can use standard cryptographic hash functions such as SHA1 [Lab95] and MD5 [Riv92]. To authenticate packet P_j, the receiver simply computes the hash of P_j and checks whether it equals the corresponding hash value carried in packet P_i. Since the cryptographic hash function in use is collision resistant, it is computationally infeasible to find a different packet P' that hashes to the same authenticated value. Therefore, the authentication of P_i enables the authentication of P_j. We call the directed acyclic graph formed by the n nodes and the edges corresponding to the authentication relationship an *authentication graph*.

Due to packet loss, each receiver may only receive a subset of the packets and hence only a subset of the vertices in the graph. We say a vertex is *up* if the corresponding packet is received, and we say a path is an *up path* if all the vertices on the path are up.

One of the packets, denoted as R, will be signed with the sender's public key using a public-key signature algorithm such as RSA [RSA78] or DSA [DSS92, Nat00]. Receivers authenticate R on arrival by verifying the digital signature in the packet. Receivers authenticate other packets by following the edges starting from R in the authentication graph. Receivers can authenticate packet P_i if and only if there is an up path from the signature packet R to P_i in the authentication graph. We denote the probability of an up path from R to P_i given that P_i is received as $\Pr[R \rightarrow P_i | P_i \text{ is received}]$. Note that we assume that all receivers receive the signature packet R. (We can increase the probability that R is received through a variety of means, including sending multiple copies of R.) We only make this assumption for the signature packet and not for any other data packets.

We want to design authentication graphs that are efficient and which allow receivers to authenticate packets with high probability. For most applications, we assume the sender and the receiver are capable of buffering a large amount of data; hence, the most important efficiency metrics are overhead per packet and authentication probability ($\Pr[R \to P_i | P_i \text{ is received}]$). Because each edge starting from vertex i in the authentication graph induces a hash value appended to packet P_i, typically at least 10 or 20 bytes overhead per packet, we would like the graph to have low constant degree independent of graph size. Also, even when packets have a high loss rate, a receiver should still be able authenticate a received packet with high probability. We would like to prove a lower bound of the authentication probability for all packets and we want this lower bound to be independent of the graph size.

It is often desirable to have low receiver authentication delay. When a receiver receives packet P_i, we do not want to wait for a large number of subsequent packets to be sent (and maybe received) before the receiver can authenticate P_i. Therefore, we sign the first packet in the stream in most of our constructions. (In section 5, we mention scenarios where we sign the last packet in a stream.)

We assume a probabilistic model for packet loss where each packet in the stream can be received with probability p independent of other packets. Perrig et al. proposed some general solutions for the authentication graph for this probabilistic model [PCTS00]. But they do not provide a proven lower bound for the authentication probability. Miner and Staddon proposed to use a ρ-random graph as the authentication graph for this probabilistic model [MS01]. In their solution, each edge between two vertices exist with probability ρ. Unfortunately this results in high degrees in the graph — many vertices in the graph have degree linear to the number of nodes in the graph. In this paper, we propose a new construction of authentication graph based on expander graphs that has constant degree and high authentication probability independent of the graph size.

2.2 Expander Graphs

An expander graph has the property that every subset of the vertices has many neighbors. Expander graphs enjoy wide use in computer science; a very incomplete list of applications includes network constructions [FFP88], sorting [AKS83, Pip87], complexity theory [Val76], cryptography [GIL$^+$90], and pseudorandomness [AKS87]. We consider two type of expanders: bipartite expanders and ordinary expander graphs. We use bipartite expanders in our construction of authentication graphs and ordinary expander graphs in our construction of overlay networks.

Definition 2.1 (bipartite graph). *A bipartite graph $G = (V_1, V_2, E)$ is an undirected graph consisting of two non-overlapping sets of vertices V_1 and V_2 and edges connecting the two sets of vertices, i.e. if an edge $(u, v) \in E$, then either $u \in V_1, v \in V_2$ or $u \in V_2, v \in V_1$. G is called a (n_1, n_2)-bipartite graph with degree (d_1, d_2) if $|V_1| = n_1, |V_2| = n_2$, and every node in V_1 has degree at most d_1, every node in V_2 has degree at most d_2. If $d_1 = d_2$ we say the degree is d_1.*

In bipartite expanders, V_1 and V_2 may have different sizes, so we expect different expansion factors on the two sides.

Definition 2.2 (bipartite expander). *A bipartite graph $G = (V_1, V_2, E)$ is (c_1, c_2)-expanding if for $i = 1, 2$, for every $S \subseteq V_i$ where $|S| \leq |V_{3-i}|/(2c_i)$, $|\Gamma(S)| \geq c_i|S|$, where $\Gamma(S)$ is the set of neighbors of S in V_{3-i}. If $c_1 = c_2$ we say the graph is c_1-expanding.*

Definition 2.3 (ordinary expander graph). *An undirected graph $G = (V, E)$ is c-expanding if for every $S \subseteq V$ where $|S| \leq |V|/(2c)$, $|\Gamma(S)| \geq (c - 1)|S|$, where $\Gamma(S)$ is the set of neighbors of S (not including S).*

Note that there are several slightly different definitions of expanders used in the literature. Also in our exposition below, we sometimes assume that some quantities are integers. This can be achieved by calculating ceilings and floors, and does not substantially change our analysis.

It is not hard to show that random graphs are almost always excellent expanders (as illustrated in Appendix A), but we can also explicitly construct constant degree expander graphs [Mar73, GG81]. While random graphs give better parameters than these deterministic constructions, there is no known way to verify such strong parameters, and pseudorandom generators used in practice may fail to give such parameters. We therefore recommend using a deterministic construction. Lubotzky, Phillips, and Sarnak [LPS88], and independently

Margulis [Mar82], describe one efficient explicit construction of expanders. Lubotzky et al. give, for every $d = p + 1$ where p is a prime congruent to 1 modulo 4, $n = q + 1$ where q is a prime congruent to 1 modulo 4, an explicit construction of a graph with n vertices and degree d, called a *Ramanujan graph.* The Ramanujan graph construction can be used to construct both bipartite expander graphs and non-directed expander graphs. Using a result by Tanner [Tan84], we have the following theorem:

Theorem 2.1. *[LPS88] The Ramanujan graph construction give a (n, n)-bipartite expander graph of degree d for every $n = q + 1$, $d = p + 1$ where p and q are two primes congruent to 1 modulo 4. These graphs are $d/8$-expanding. The same construction can be used to construct ordinary expander graphs with n vertices and degree d and $d/8$-expanding.*

In our analysis we will also use the Chernoff bound:

Theorem 2.2. *[Che52] Let $X_1, X_2, \ldots, X_n$ be independent random variables such that, for $1 \leq i \leq n$, $\Pr[X_i = 1] = p_i, \Pr[X_i = 0] = 1-p_i$, where $0 < p_i < 1$. Define $X = \sum_1^n X_i$, and define $\mu = E[X]$. Then for $0 < \delta \leq 1$, $\Pr[X < (1-\delta)\mu] < \exp(-\mu\delta^2/2)$.*

From the Chernoff bound, we can easily obtain the following corollary:

Corollary 2.3. *Given a set of s nodes where each node is up independently with probability p, the probability that at least $ps/2$ nodes are up is at least $(1 - \exp(-ps/8))$.*

3 Expander-Based Authentication: Construction and Analysis

3.1 Construction and Analysis of DAG Expanders

We use the expansion property of expanders to construct an authentication graph allowing a receiver to authenticate a received packet with high probability. Because an authentication graph is a directed acyclic graph (DAG) rooted at the signature packet, we cannot directly use existing bipartite or ordinary expander constructions. In this section, we propose a new construction to build a directed acyclic expander graph, a *DAG expander.* Our DAG expander is a DAG rooted at the signature packet with several levels. We put edges between two neighboring levels in the tree using the bipartite expander graph we construct below.

We first use a (n, n)-bipartite expander graph with degree d and expansion factor c to construct a $(n/a, n)$-bipartite expander:

Lemma 3.1. *Given a (n,n)-bipartite expander graph with degree d and expansion factor c, we can explicitly construct a $(\frac{n}{a}, n)$-bipartite expander of degree (da, d) and is $(ac, \frac{c}{a})$-expanding.*

Proof. Suppose we are given a (n, n) bipartite graph $G' = (V_1', V_2, E')$ of degree d and expansion factor c. Label the vertices in V_1' as $v_0', v_1', \ldots, v_{n-1}'$. We form a new graph $G = (V_1, V_2, E)$ by contracting V_1' a vertices at a time. In other words, we merge the vertices $v_{a\cdot i}', \ldots, v_{a\cdot i+a-1}'$ into one vertex v_i. Thus, $V_1 = \{v_i\}_{0 \leq i < n/a}$, where the neighbors of v_i in G are all the neighbors of $v_{a\cdot i}', \ldots, v_{a\cdot i+a-1}'$ in G'.

To see the expansion properties, consider any $X \subset V_1$ where $|X| \leq \frac{n}{2ac}$. The neighbors of X are the neighbors of the corresponding set X' in V_1', where $|X'| = a|X|$. Thus, $|\Gamma(X)| = |\Gamma(X')| \geq c|X'| = ac|X|$. Now consider any $Y \subset V_2$. Let $\Gamma'(Y)$ denote the set of neighbors of Y in V_1'. When $|Y| \leq \frac{n}{2c}$, $\frac{|\Gamma'(Y)|}{|Y|} \geq c$. Now $\Gamma'(Y)$ is mapped to at least $|\Gamma'(Y)|/a$ distinct vertices in V_1. Therefore for any $Y \subset V_2$ where $|Y| \leq \frac{n}{2c}$, $\frac{|\Gamma(Y)|}{|Y|} \geq \frac{c}{a}$. □

We only use the expansion from V_2 to V_1 in our construction.

Construction 3.2 (DAG Expander). *We construct a layered DAG expander using the $(\frac{n}{a}, n)$-bipartite expanders found by applying Lemma 3.1 to any (n, n)-bipartite expander graph. The zeroth layer contains the root R, and for all i the ith layer contains a^i vertices. Layers $i-1$ and i are connected using a copy of an (a^{i-1}, a^i)-bipartite expander graph from Lemma 3.1. The edges point from layer $i-1$ to layer i. Let c denote the expansion factor from the ith layer to $i-1$th layer. Figure 1 shows an example of the layer construction.*

Recall that we say a node is *up* if the corresponding packet is received by the receiver, we call a path an *up path* if all the vertices on the path are up. We first show that our construction ensures that an up node can be reached from the root R via an up path with high probability.

Theorem 3.3. *Assume each vertex except the root R in our DAG expander is up independently with probability p, where c is the expansion factor from ith layer to $i-1$th layer, $c > 4/p$ and $a > 4/p$. If a vertex v is up, then there exists an up path from*

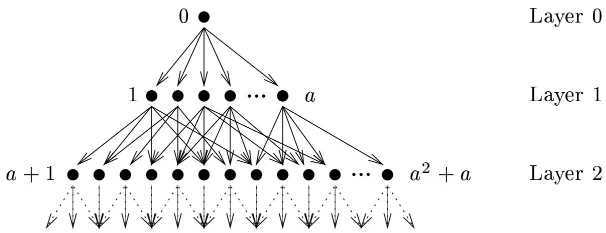

Figure 1. An example of our layer construction of the DAG expander

R to v with probability at least $1 - \frac{\exp(-cp/8)}{1-\exp(-cp/8)} - \frac{\exp(-ap/16)}{1-\exp(-ap/16)}$.

Proof. Assume vertex v is in layer t. Let S_i denote the set of vertices in layer i that are up and can reach v via up paths. The theorem follows from the following three claims.

Claim 1: Suppose $(cp/2)^{t-i-1} \leq a^i/(2c)$. Then the probability that $|S_i| \geq (cp/2)^{t-i}$ is at least

$$1 - (\sum_{\ell=1}^{t-i} \exp(-\frac{1}{4} \cdot (\frac{cp}{2})^{\ell})) \geq 1 - \frac{\exp(-cp/8)}{1-\exp(-cp/8)}.$$

Proof of Claim 1: We prove this by induction from $i = t-1$ down to 0. It is trivially true for $i = t-1$ due to the Chernoff bound. Suppose it is true for $i+1$, so with high probability $|S_{i+1}| \geq s = (cp/2)^{t-i-1}$, and $s \leq a^i/(2c)$. From the expanding property, these vertices are adjacent to at least $c \cdot s$ vertices in layer i. The probability that at least $\frac{p}{2} \cdot c \cdot s = (\frac{cp}{2})^{t-i}$ of these vertices are up is at least $1 - \exp(-\frac{1}{4} \cdot (\frac{cp}{2})^{t-i})$ from the Chernoff bound (Corollary 2.3).

Therefore, if $(cp/2)^{t-i-1} \leq a^i/(2c)$, then for all j such that $i < j \leq t$, $(cp/2)^{t-j} \leq a^{j-1}/(2c)$, and the probability that $|S_i| \geq (cp/2)^{t-i}$ is at least

$$\begin{aligned} 1 - (\sum_{\ell=1}^{t-i} \exp(-\frac{1}{4} \cdot (\frac{cp}{2})^{\ell})) &\geq 1 - (\sum_{\ell=1}^{t-i} \exp(-\frac{\ell cp}{8})) \\ &\geq 1 - \frac{\exp(-cp/8)}{1-\exp(-cp/8)} \end{aligned}$$

Claim 2: With probability at least $1 - \frac{\exp(-cp/8)}{1-\exp(-cp/8)}$, there is an m for which $|S_m| \geq a^{m-1}/(2c)$.

Proof of Claim 2: Let m be the largest integer such that $(cp/2)^{t-m} \geq a^{m-1}/(2c)$. Then $(cp/2)^{t-m-1} < a^m/(2c)$. From Claim 1, $|S_m| \geq (cp/2)^{t-m}$ with probability at least $1 - \frac{\exp(-cp/8)}{1-\exp(-cp/8)}$.From $(cp/2)^{t-m} \geq a^{m-1}/(2c)$, we get $|S_m| \geq a^{m-1}/(2c)$ with probability at least $1 - \frac{\exp(-cp/8)}{1-\exp(-cp/8)}$.

Claim 3: If $|S_m| \geq a^{m-1}/(2c)$, then $|S_i| \geq pa^i/4$ for all $i < m$, with probability at least

$$1 - \sum_{i=1}^{m-1} (\exp(-a^i p/8)) \geq 1 - \frac{\exp(-ap/16)}{1-\exp(-ap/16)}.$$

Proof of Claim 3: We proceed by induction from $i = m-1$ down to $i = 0$.

When $i = m-1$, we have $|S_m| \geq a^{m-1}/(2c)$. By the expanding property, the number of vertices in layer $m-1$ that are adjacent to the vertices in S_m is at least $c \cdot (a^{m-1}/(2c)) = a^{m-1}/2$. By the Chernoff bound, with probability at least $1 - \exp(-a^{m-1}p/16)$, the fraction of vertices that are up will be at least $p/2$.

Suppose claim holds for $i+1$, that is, with high probability $|S_{i+1}| \geq s = pa^{i+1}/4 \geq a^i/(2c)$. By the expanding property, the number of vertices adjacent to these up vertices is at least $c(a^i/(2c)) = a^i/2$. Again we apply the Chernoff bound to find that with probability at least $1 - \exp(-a^i p/16)$, the fraction of adjacent vertices that are up is at least $p/2$ (in layer i, $pa^i/4$ vertices). If we are given $|S_{i+1}| \geq pa^{i+1}/4$, then we have $|S_i| \geq pa^i/4$ with probability at least $1 - \exp(-a^i p/16)$.

So given $|S_m| \geq a^{m-1}/(2c)$, we have $|S_i| \geq pa^i/4$ for all $i < m$, with probability at least

$$1 - \sum_{i=1}^{m-1} (\exp(-a^i p/16)) \geq 1 - \frac{\exp(-ap/16)}{1-\exp(-ap/16)}.$$

Now that we have shown the three claims, we can see the proof of Theorem 3.3. Recall that by hypothesis $pa/4 > 1$. So, combining the three claims, we have $|S_1| \geq pa/4 > 1$ with probability at least

$$1 - \frac{\exp(-cp/8)}{1-\exp(-cp/8)} - \frac{\exp(-ap/16)}{1-\exp(-ap/16)}.$$

□

3.2 Expander-based Authentication Graph

We use our DAG expander construction to form the authentication graph. In particular, let the root R be the first packet P_0. P_0 is digitally signed and, by assumption, will reach all receivers. We number the vertices from 0 to $n-1$ layer by layer.

Any vertex on layer i has a lower number than any vertex on layer $i+1$. Let vertex i correspond to packet P_i. In this authentication graph, each packet except for packets corresponding to leaves on the DAG expander has a constant number da embedded hash values. Note that not only is da constant, it is independent of the size of the graph. The authentication probability is at least $1 - \frac{\exp(-cp/8)}{1-\exp(-cp/8)} - \frac{\exp(-ap/16)}{1-\exp(-ap/16)}$.

Take a degree d Ramanujan expander (Theorem 2.1). This Ramanujan expander will have an expansion factor of at least $d/8$. Applying Construction 3.2 we get a *DAG expander.* We immediately have:

Corollary 3.4. *Assume we have a DAG expander. Assume each vertex in the DAG expander except the root R is up independently with probability p, and $d > 32a/p$ and $a > 4/p$. If a vertex v is up, then there exists an up path from R to v with probability at least* $1 - \frac{\exp(-dp/(64a))}{1-\exp(-dp/(64a))} - \frac{\exp(-ap/16)}{1-\exp(-ap/16)}$.

Therefore, given a threshold of accepted authentication probability ϵ, the estimated packet arrival probability p, we can select parameter a and the lowest value d to ensure that the authentication probability is above ϵ.

4 Expander-based Overlay Networks: Construction and Analysis

Recall from Section 1.2 that an overlay network is formed from a subset of nodes drawn from an underlying network. Participating nodes communicate via *virtual links* between two nodes that may not be directly connected in the underlying network. These virtual links form the topology of the overlay network. We assume that virtual links are reliable but that nodes may join and leave the network dynamically. We say a node is *up* if it is currently operating in the overlay network, otherwise, we say the node is *down.* And we say a path is an *up path* if all the nodes on the path are up.

Using undirected expander graphs, we construct an overlay network with each node having constant degree independent of the size of the graph. Assume each node in the overlay is up independently with probability p. We find a lower bound on the probability that between two nodes there is a short up path of length $O(\log n)$. This lower bound is independent of the number of nodes in the overlay. Because we have constant degree of the nodes and short up paths, our construction yields particularly efficient overlay networks.

It is particularly interesting to consider broadcasts in overlay networks. Because of the properties described above, broadcast in these networks has low overhead compared to networks with high degree or large diameter. We apply broadcast in our overlay networks to a computer security problem. In Section 1.2, we discuss an efficient protocol for decentralized certificate revocation. Our protocol can be used to distribute public key revocation lists. Our protocol improves on previous results [WLM00] by being highly scalable.

4.1 Construction and Analysis of Overlay Networks

Given n nodes, we construct the overlay network using an explicit expander graph construction. For simplicity, we show the analysis of our construction using the Ramanujan expander. The results can be easily generalized to other expander constructions.

Given n nodes, we form the overlay network as the Ramanujan expander graph with n nodes and degree d. Each node in the graph corresponds to a host in the overlay network, and each edge in the graph indicates the virtual link between the two connected hosts in the overlay network. When a node wants to broadcast a message to all other nodes in the overlay, it sends the message to all its neighbors. When a node receives the message from a neighbor, it forwards the message to all of its other neighbors unless it has already seen the message before.[1]. Assuming transmission time over each virtual link is bounded, the latency of the transmission over the path of virtual links is bounded by a constant factor of the length of the shortest up path between the receiving and the sending node. We prove that a path exists with high probability and that the path is short.

Theorem 4.1. *Let G be an undirected Ramanujan expander graph on n nodes with degree d. Assume each node in the graph is up independently with probability p. For any two up nodes v and w, the probability that there is an up path of length $O(\log n)$ from v to w is* $1 - \frac{2\exp(-dp/64)}{1-\exp(-dp/64)}$, *given that $d \geq (8/p)^2$. Similarly, a broadcast message by v*

[1] Discarding duplicate messages reduces the number of broadcasts and prevents cycles of message forwarding. Receivers can detect duplicate messages in several ways. For example, each node can store the hash values of messages it recently forwarded and each message could contain a expiration time. The node can check whether the packet is a duplicate by comparing it against the table of its recently forwarded hash values. If the message is expired, the node simply drops it.

will reach a particular node in an up path of length $O(\log n)$ with probability at least $1 - \frac{2\exp(-dp/64)}{1-\exp(-dp/64)}$.

The theorem follows from the following two lemmas.

Lemma 4.2. *With probability at least $1 - \frac{\exp(-dp/64)}{1-\exp(-dp/64)}$, any up node v can reach more than $pn/4$ up nodes within distance $O(\log n)$ via up paths.*

Proof. Let S_i denote the number of up nodes that v can reach via up paths of at most length i. The lemma follows from the following two claims.

Claim 1: Suppose $(dp/16)^{i-1} \leq 4n/d$. Then the probability that $|S_i| \geq (dp/16)^i$ is at least

$$1 - (\sum_{\ell=1}^{i} \exp(-\frac{1}{4} \cdot (\frac{dp}{16})^{\ell})) \geq 1 - \frac{\exp(-dp/64)}{1-\exp(-dp/64)}.$$

Proof of Claim 1: The proof here is similar to the proof in Claim 1 for Theorem 3.3. We prove this by induction from $i = 0$. It is trivially true for $i = 0$. Suppose it is true for i, so with high probability $|S_i| \geq s = (dp/16)^i$ and $s \leq 4n/d$. From the expanding property, these vertices are adjacent to at least $\frac{d}{8} \cdot s$ vertices (including the vertices already in S_i). The probability that at least $\frac{p}{2} \cdot \frac{d}{8} \cdot s = (\frac{dp}{16})^{i+1}$ of these vertices are up is at least $1 - \exp(-\frac{1}{4} \cdot (\frac{dp}{16})^{i+1})$ from the Chernoff bound. So $|S_{i+1} \geq (dp/16)^{i+1}$ with probability at least

$$1 - (\sum_{\ell=1}^{i+1} \exp(-\frac{1}{4} \cdot (\frac{dp}{16})^{\ell})) \geq 1 - \frac{\exp(-dp/64)}{1-\exp(-dp/64)}.$$

Claim 2: With probability at least $1 - \frac{\exp(-dp/64)}{1-\exp(-dp/64)}$, there is an m for which $|S_m| \geq 4n/d$ and $m = O(\log n)$.

Proof of Claim 2: Let m be the smallest integer such that $(dp/16)^m \geq 4n/d$. Then $(dp/16)^{m-1} < 4n/d$. From Claim 1, it immediately follows that $|S_m| \geq (dp/16)^m \geq 4n/d$ with probability at least $1 - \frac{\exp(-dp/64)}{1-\exp(-dp/64)}$. □

Lemma 4.3. *Any two sets of size at least $2n/\sqrt{d}$ in a Ramanujan expander with n nodes and degree d have at least one edge between the two sets.*

Proof. This follows from Lemma 2.4 in Chapter 9 of [ASE92]. □

Because $d \geq (8/p)^2$, we have $pn/4 \geq 2n/\sqrt{d}$. From Lemma 4.2 and 4.3, we conclude that any two up nodes can be reached from each other via up path of length $O(\log n)$ with probability at least $1 - \frac{2\exp(-dp/64)}{1-\exp(-dp/64)}$.

4.2 Application to Decentralized Certificate Revocation

In Section 1.2, we discussed the Wright, Lincoln, and Millen decentralized model for distributing certificate revocations [WLM00]. By propagating messages in their *depender graph*, they hoped to distribute revocation lists to all (or most) nodes. Their construction uses a graph of degree k and can tolerate $k-1$ node failures. However, if each node has an independent failure probability p, as the number of nodes in the network increases, the number of expected failures will increase, and the degree of the graph will increase. In fact, the degree of the Wright et al. depender graph increases linearly with the number of nodes in the network. This means that each node will need to send out more messages. As the size of the network becomes large, this will introduce substantial delays in distributing revocation lists. Further examination of their protocol shows that the number of revocation messages received by each node will also grow linearly.

If we use the overlay network constructed in the previous section, we can have an effective graph for distributing certificate revocation messages. The graph will have constant degree, and the number of revocation messages sent (or received) is at most the degree of the graph. Furthermore, with high probability, each node is reachable by an up path of length $O(\log n)$. So even if a high fraction of nodes fails, each up node will receive the revocation message in $O(\log n)$ steps with probability $1 - \frac{2\exp(-dp/64)}{1-\exp(-dp/64)}$.

4.3 Survivable networks

This result may bear on an important open question in computer security: how can we make networks survivable against directed attacks. We consider an adversary who can attack individual nodes in our network. The question of survivable networks is important at many levels. For example, it has clear implications for protecting national infrastructure in the face of hostile attacks. It also has implications for applications that may be unpopular with some Internet users. For example, it appears that much of the material being exchanged over some peer-to-peer file sharing systems (such as Gnutella) is protected by copyright. In this case, copyright holders have an interest in seeing the Gnutella overlay network disrupted. In contrast, the users of Gnutella want to prevent disruption of the Gnutella overlay network.

Depending on the application and nature of the adversary, we may be interested in using our expander graph constructions to build a survivable overlay network (for example, to prevent disruptions to an overlay network such as Gnutella) or a survivable underlying network (for example, to build a highly survivable Internet) or some combination.

How many nodes can an adversary successfully attack? If the adversary can only successfully attack a small number of nodes, then it is easy to consider a variety of techniques that can protect those nodes. If the adversary can successfully attack all nodes in the network, we clearly have a lost cause and communication will be completely disrupted. Perhaps the most interesting case to consider is an adversary who can attack a constant proportion of nodes in our network. If an adversary can take out nodes in the network with independent probability, then the results described above immediately apply and provide an outline of how network designers can build highly survivable.

What about an adversary who can take out specific nodes in the network? If such an attacker knows the topology of our overlay network, he could try to isolate certain nodes. It is an open problem to determine how many nodes an adversary could isolate. (If it is possible to disguise the topology of a network from an attacker, it may be possible to keep an adversary from knowing which nodes to attack.)

5 Related Work

5.1 Expander Graphs

We have shown how to use expanders to construct authentication graphs and overlay networks. Our analysis is based on the expansion property of explicit expander constructions. Explicit expander construction is still an active area of research in graph theory. More efficient expander constructions, such as [CRVW02], may improve the efficiency of our construction. Our analysis still allows some room for improvements. For example, we can improve our probability bound and reduce d by using Kahale's result showing that small sets in Ramanujan graphs have expansion close to $d/2$ [Kah95]. Researchers have studied applying expander graphs to certain networking problems. For example, Broder et al. investigate the problem of virtual circuit switching using expander graphs [BFU97], Peleg and Upfal studied the problem of constructing disjoint paths on expander graphs [PU89].

5.2 Stream Authentication

Many researchers have studied the problem of efficient authentication of digital streams. Gennaro and Rohatgi [GR97] propose a model in which the sender signs the first packet and inserts the hash of each block into the preceeding block. Their solution does not tolerate packet loss. Rohatgi later proposes to use k-time signatures for stream authentication but the scheme still requires that each receiver receives at least one out of every k packets [Roh99].

Wong and Lam propose a tree-based authentication scheme which amortizes one digital signature over n packets. Their scheme adds one digital signature and $O(\log n)$ hash values to each packet [WL98].

Canetti et al. construct a solution using k different keys to authenticate every message with k different MAC's [CGI$^+$99]. Their solution is only secure when the number of colluding members is less than k.

Anderson et al. propose a scheme which provides stream authentication between two parties and does not tolerate packet loss [ABC$^+$98]. Other researchers extend the approach of Anderson et al.

Perrig et al. propose the TESLA protocol which can tolerate packet loss but requires loose time synchronization [PCTS00, PCST01].

Perrig et al. also propose the first general form of graph-based authentication with constant degree, EMSS [PCTS00]. In EMSS, the hash value of a packet P_i will be embedded into a constant number of other packets, where the pattern of hash embedding is chosen either in priori or randomly. They mainly give simulation results and do not prove a lower bound on the probability that a received packet can be authenticated.

Golle and Modadugu propose a hash-based scheme that can only tolerate a single burst of loss [GM01].

Miner and Staddon generalize the approach by Perrig et al.[PCTS00] and propose p-random graphs for graph authentication [MS01]. They gave a theoretical bound on the authentication probability which relies on p. In their scheme, a large fraction of the packets carry $\Theta(pn)$ hash values, where n is the number of packets in the stream, making their approach unscalable. They also propose to differentiate packets according to their level of importance and provide a higher probability of authentication to more important packets to reduce overhead. Similar techniques can be applied in our construction as well.

A number of previously cited researchers note that if the sender does not want to buffer data, we can reverse all the links in our authentication graph and put the signature packet at the end. In this way, the sender does not need to buffer packets but the receiver has to buffer packets until it gets the signature packet before it can authenticate packets. This is possible with our approach also. But we do not think it is a good idea for our system or other systems. As several researchers have noted, reversing links makes protocols susceptible to denial-of-service attack and is in general not recommended. Note that our solution also provides non-repudiation.

5.3 Overlay Networks

Many overlay networks have been proposed [CRZ00, CMB00, Fra, JGJ+00]. Some are only for small groups. Most previous work is simulation based and does not provide any proven lower bound on the probability of reachability.

5.4 Certificate Revocation

Many researchers have discussed issues on certificate revocation and proposed various techniques for improving efficiency of revocation (an incomplete list of recent work including [Riv98, NN98, Mye98, MR00, MJ00, Koc98, KAN99, FL98, Coo99]). Most previous work in certificate revocation focused on a centralized model where a key server is responsible for maintaining and distributing the certificate revocation lists (CRLs). Wright, Lincoln and Millen recently proposed a decentralized model for certificate revocation [WLM00] that we discuss in Section 1.2.

6 Conclusion

We propose a new construction based on expander graphs to authenticate long digital streams over lossy networks. Our construction is efficient in the sense that the authentication graph has constant degree, a major improvement over the previous work that uses $\Theta(n)$ degree, where n is the number of nodes in the graph. Our construction also enables a high probability of authentication. In particular, we provide a proven lower bound of the probability that a packet can be authenticated upon arrival; our lower bound is independent of the size of the graph. We apply our analysis techniques to undirected expander graphs, and our results can be used to construct efficient, robust, and scalable overlay networks. We use our overlay network construction to provide a more efficient solution to the decentralized certification revocation problem.

Expander graphs are a powerful yet relatively new tool. We hope that our analysis and application of expander graphs can provide new insight in solving related problems.

References

[6Bo] 6Bone. `http://www.6bone.net`.

[ABC+98] R. Anderson, F. Bergadano, B. Crispo, J. Lee, C. Manifavas, and R. Needham. A new Family of Authentication Protocols. *Operating Systems Review*, 1998.

[ABo] ABone. `http://www.isi.edu/abone`.

[AKS83] M. Ajtai, J. Komlós, and E. Szemerédi. Sorting in $c \log n$ Parallel Steps. *Combinatorica*, 3:1–19, 1983.

[AKS87] M. Ajtai, J. Komlós, and E. Szemerédi. Deterministic Simulation in Logspace. In *ACM Symposium on Theory of Computing*, 1987.

[ASE92] N. Alon, J. H. Spencer, and P. Erdős. *The Probabilistic Method.* Wiley–Interscience Series, John Wiley & Sons, Inc., New York, 1992.

[BFU97] A. Z. Broder, A. M. Frieze, and E. Upfal. Static and dynamic path selection on expander graphs: a random walk approach. In *ACM Symp. on Theory of Computing*, 1997.

[CGI+99] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. Multicast Security: A Taxonomy and Some Efficient Constructions. In *IEEE INFOCOM*, March 1999.

[Che52] H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.

[CMB00] Y. Chawathe, S. McCanne, and E. Brewer. An Architecture for Internet Content Distribution as an Infrastructure Service. http://yatin.chawath.com/papers/scattercast.ps, 2000.

[Coo99] D. Cooper. A Model of Certificate Revocation. In *Computer Security Applications Conference*, 1999.

[CRVW02] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. Randomness conductors and constant-degree expansion beyond the degree 2 barrier. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, 2002.

[CRZ00] Y. Chu, S. Rao, and H. Zhang. A Case for End System Multicast. In *ACM SIGMETRICS*, 2000.

[DSS92] The Digital Signature Standard Proposed by NIST. *Communications of the ACM*, 35(7):36–40, July 1992.

[Eri94] H. Eriksson. MBone: The Multicast Backbone. *Communications of the ACM*, 37, August 1994.

[FFP88] P. Feldman, J. Friedman, and N. Pippenger. Wide-Sense Nonblocking Networks. *SIAM Journal on Discrete Mathematics*, 1:158–173, 1988.

[FL98] B. Fox and B. LaMacchia. Certificate Revocation: Mechanics and Meaning. In *Financial Cryptography*, 1998.

[Fra] P. Francis. Yoid: Your Own Internet Distribution. http://www.aciri.org/yoid.

[GG81] O. Gabber and Z. Galil. Explicit Construction of Linear Sized Superconcentrators. *Journal of Computer and System Sciences*, 22:407–420, 1981.

[GIL+90] O. Goldreich, R. Impagliazzo, L. Levin, R. Venkatesan, and D. Zuckerman. Security preserving amplification of hardness. In *IEEE Symposium on Foundations of Computer Science*, pages 318–326, 1990.

[GM01] P. Golle and N. Modadugu. Authenticating Streamed Data in the Presence of Random Packet Loss. In *Network and Distributed Systems Security (NDSS 2001)*, February 2001.

[Gnu] http://www.gnutella.com.

[GR97] R. Gennaro and P. Rohatgi. How to Sign Digital Streams. In *Advances in Cryptology – CRYPTO '97*, pages 180–197, 1997.

[JGJ+00] J. Jannoti, D. Gifford, K. Johnson, F. Kasshoek, and J. Toole. Overcast: Reliable multicasting with an overlay network. In *USENIX Symposium on Operating Systems Design and Implementation*, 2000.

[Kah95] N. Kahale. Eigenvalues and Expansion of Regular Graphs. *Journal of the ACM*, 42:1091–1106, 1995.

[KAN99] H. Kikuchi, K. Abe, and S. Nakanishi. Performance Evaluation of Certificate Revocation Using k-Valued Hash Tree. In *Information Security Workshop*, 1999.

[Koc98] P. Kocher. On Certificate Revocation and Validation. In *Financial Cryptography'98*, 1998.

[Lab95] National Institute of Standards and Technology (NIST)(Computer Systems Laboratory). Secure Hash Standard. Federal Information Processing Standards Publication FIPS PUB 180-1, April 1995.

[LPS88] A. Lubotzky, R. Philips, and P. Sarnak. Ramanujan Graphs. *Combinatorica*, 8:261–277, 1988.

[Mar73] G.A. Margulis. Explicit Construction of Concentrators. *Problems of Information Transmission*, 9:325–332, 1973.

[Mar82] G. Margulis. Explicit Constructions of Graphs without Short Cycles and Low-density Codes. *Combinatorica*, 2(1), 1982.

[MJ00] P. McDaniel and S. Jamin. Windowed Certificate Revocation. In *IEEE INFOCOM*, 2000.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms.* MIT Press, 1995.

[MR00] P. McDaniel and A. Rubin. A Response to 'Can We Eliminate Certificate Revocation Lists?'. In *Financial Cryptography*, 2000.

[MS01] S. Miner and J. Staddon. Graph-Based Authentication of Digital Streams. In *IEEE Symposium on Research in Security and Privacy*, pages 232–246, May 2001.

[Mye98] M. Myers. Revocation: Options and Challenges. In *Financial Cryptography*, 1998.

[Nat00] National Institute of Standards and Technology (NIST). The Digital Signature Standard (DSS). FIPS PUB 186-2, January 2000.

[NN98] M. Naor and K. Nissim. Certificate revocation and certificate update. In *Proc. 7th USENIX Security Symposium*, 1998.

[PCST01] Adrian Perrig, Ran Canetti, Dawn Song, and J. D. Tygar. Efficient and Secure Source Authentication for Multicast. In *Network and Distributed System Security Symposium, NDSS '01*, pages 35–46, February 2001.

[PCTS00] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. Efficient Authentication and Signature of Multicast Streams over Lossy Channels. In *IEEE Symposium on Research in Security and Privacy*, pages 56–73, May 2000.

[Pip87] N. Pippenger. Sorting and Selecting in Rounds. *SIAM Journal on Computing*, 16(6):1032–1038, December 1987.

[PU89] D. Peleg and E. Upfal. Constructing disjoint paths on expander graphs. *Combinatorica*, 9, 1989.

[Riv92] R. Rivest. The MD5 message-digest algorithm. Internet Request for Comment RFC 1321, Internet Engineering Task Force, April 1992.

[Riv98] R. Rivest. Can we Eliminate Certificate Revocation Lists? In *Financial Cryptography*, 1998.

[Roh99] P. Rohatgi. A Compact and Fast Hybrid Signature Scheme for Multicast Packet. In *ACM Conference on Computer and Communications Security*, pages 93–100, November 1999.

[RSA78] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[Tan84] R. Tanner. Explicit Construction of Concentrators from Generalized n-gons. In *SIAM Journal on Algebraic and Discrete Methods*, 1984.

[Val76] L. Valiant. Graph Theoretic Properties in Computational Complexity. *Journal of Computer and System Sciences*, 13:278–285, 1976.

[WL98] C. Wong and S. Lam. Digital Signatures for Flows and Multicasts. In *IEEE International Conference on Network Protocols*, 1998.

[WLM00] R. Wright, P. Lincoln, and J. Millen. Efficient Fault-Tolerant Certificate Revocation. In *ACM Conference on Computer and Communications Security*, November 2000.

[Zim95] P. Zimmermann. *The Official PGP User's Guide*. MIT press, 1995.

A Probability of Random Graphs Being Expanders

As usual $[n]$ denotes the set $\{1, 2, \dots, n\}$.

Here's a modified definition of bipartite expander for ease of explaination.

Definition A.1 (expander). *A bipartite graph $G = (V, W, E)$ is (k, c)-expanding if for every $S \subseteq V$ of cardinality at most k, $|\Gamma(S)| \geq c|S|$, where $\Gamma(S)$ is the set of neighbors of S.*

Proposition A.1. *Create a random bipartite graph G on $[n] \cup [m]$ by choosing, for each vertex in $[n]$, d random neighbors in $[m]$ (without replacement). Assume that $ce^d \frac{n}{m} \left(\frac{ck}{m}\right)^{d-1-c} \leq 1/2$. Then the probability that G is not (k, c)-expanding is less than*

$$2\left(ce^{c+1}\frac{n}{m}\left(\frac{2c}{m}\right)^{d-1-c}\right)^2$$

Proof. Consider sets of size $s \leq k$. The probability that a fixed set of size s has all its neighbors in a particular set of size cs is at most $(cs/m)^{ds}$. Taking the union over all subsets of $[n]$ of size s and all subsets of $[m]$ of size cs, the probability that any set of size s fails to expand is at most

$$\begin{aligned}\binom{n}{s}\binom{m}{cs}\left(\frac{cs}{m}\right)^{ds} &\leq \left(\frac{en}{s}\right)^{s}\left(\frac{em}{cs}\right)^{cs}\left(\frac{cs}{m}\right)^{ds} \\ &= \left(e^{c+1}\frac{cn}{m}\left(\frac{cs}{m}\right)^{d-1-c}\right)^{s}\end{aligned}$$

Denoting this last expression by p_s, we see that

$$\begin{aligned}p_s/p_{s-1} &= ce^{c+1}\frac{n}{m}\left(\frac{c}{m}\right)^{d-1-c}\left(\frac{s^s}{(s-1)^{s-1}}\right)^{d-c-1} \\ &\leq ce^{d}\frac{n}{m}\left(\frac{cs}{m}\right)^{d-1-c}\end{aligned}$$

Therefore, using $ce^d\frac{n}{m}\left(\frac{ck}{m}\right)^{d-1-c} \leq 1/2$, the probability of G not expanding is at most

$$\sum_{s=2}^{k} p_s \leq p_2\sum_{s=1}^{k} 2^{1-s} < 2p_2,$$

as required.

□